Connexité et analyse des données non linéaires

Catherine Aaron

UNIVERSITE DE PARIS I | PANTHEON SORBONNE

THESE

pour obtenir
le TITRE de DOCTEUR EN SCIENCES
SPECIALITE : MATHEMATIQUES APPLIQUEES

TITRE

CONNEXITE ET ANALYSE DES DONNEES NON LINEAIRES

Par

Catherine AARON

Soutenue le 15 decembre 2005

Jury :

Marie	COTTRELL	*Directeur*
Richard	EMILION	*Examinateur*
Jean-Claude	FORT	*Directeur*
Barbara	HAMMER	*Rapporteur*
Samuel	KASKI	*Examinateur*
Ludovic	LEBART	*Rapporteur*
Manuel	SAMUELIDES	*Rapporteur*

Connexité et Analyse des données non linéaires

Connexité et Analyse des données non linéaires

Connexité et Analyse des données non linéaires

9th January 2006

REMERCIEMENTS

Pour leur patience, leur disponibilité, leur encadrement, leur accueil, leur humour, leur bonne humeur et les discussions endiablées sur les meilleurs sites à champignons (et la cuisson du magret), je voudrais remercier Marie Cottrell et Jean-Claude Fort. Je voudrais aussi les remercier pour avoir accepté ma candidature en thèse et ce, malgré un parcours "atypique" et un manque de financement. Enfin merci pour les très longues heures de relecture qui ont permis d'éradiquer un bon nombre de fautes d'orthographe.

Pour avoir accepté d'être rapporteurs, pour avoir lu et fait des remarques pertinentes et constructives je remercie, Barbara Hammer (qui aura de plus fait l'effort de lire une thèse en français), Ludovic Lebart et Manuel Samuelides. J'adresse aussi tous mes remerciements aux membres du jury.

Je remercie toute l'équipe du Samos pour leur accueil chaleureux, leur aide mais aussi pour avoir ri à mes blagues, avoir supporté mes sauts d'humeur (notamment ces derniers temps), mes séance d'hélicoptère sur ma chaise, et surtout mes chansons fredonnées terriblement faux. Je les remercie aussi pour avoir supporté (plus ou moins bien) ma tentative de reconversion du Samos en laboratoire de biologie avec une première expérience de culture de champignons et autres bactéries en tasse.

Je remercie enfin mes parents pour ne pas avoir succombé à une crise cardiaque le jour où je leur ai annoncé ma volonté de quitter une douillette situation financière pour rejoindre le monde de la recherche et aussi pour tout leur soutien et leur affection, ma famille au complet qui aura très bien fait semblant de comprendre mes explications vaseuses lorsque je leur racontait le contenue de ma thèse, et enfin, mes amis qui m'ont soutenue (et parfois nourrie) tout au long de ces années et qui ne m'ont (pas trop) pris pour une folle d'avoir choisi une telle voie.

à ma famille, à Matthieu et à Poan-Poan-la-tête-dure

TABLE OF CONTENTS

……

INTRODUCTION : Connexité et analyse des données

On s'intéresse dans cette thèse, à la mise en évidence des propriétés de connexité dans les données à analyser. Dans le cas de l'analyse des données "classique" (i.e. linéaire), comme les surfaces de séparation des classes sont des hyperplans (des droites en dimension 2), la notion topologique sous-jacente est presque toujours la convexité. Au contraire dans tout ce qui suit, on cherche en priorité à segmenter les données en sous-ensembles (classes) connexes. Par exemple, on cherche à retrouver pour les deux exemples de la figure 0.1 d'une part les deux cercles concentriques, d'autre part les deux "U".

Fig. 0.1: Deux exemples d'ensembles ayant chacun deux classes connexes, non convexes, et non linéairement séparables

Un des intérêts de ce type de classification est de fournir des sous-ensembles connexes comme domaines de résolution d'équations différentielles ou comme domaines de définition de certaines modélisations. On verra en partie II qu'une telle classification en classes connexes est un préalable à beaucoup de méthodes d'analyse des données.

Rappelons tout d'abord les principales définitions liées à la connexité. Dans tout ce qui suit, l'ensemble des données est noté X et est une partie de $\mathbb{R}^p$, muni de la distance euclidienne d, et de la topologie associée.

Définition 0.0.1 (Connexité): L'ensemble X est connexe, si et seulement si l'une des quatre propriétés suivantes est vérifiée :

i) si X est union de deux ouverts disjoints, alors l'un vaut X et l'autre est l'ensemble vide,

ii) si X est union de deux fermés disjoints, alors l'un vaut X et l'autre est l'ensemble vide,

iii) toute fonction continue de X dans $\{0, 1\}$ est constante,

iv) les seuls sous-ensembles ouverts et fermés de X sont l'ensemble vide et X.

On utilise souvent la connexité par arc qui implique la connexité (mais la réciproque est fausse).

Définition 0.0.2 (connexité par arc): L'ensemble X est connexe par arc si et seulement si :

$$\forall (x, y) \in X^2,\ \exists u \in \mathcal{C}^0([0, 1], X) \text{ telle que } u(0) = x \text{ et } u(1) = y$$

c'est-à-dire si et seulement si il existe un chemin continu liant x à y.

Proposition 0.0.1 (décomposition en composantes connexes maximale): Tout ensemble X de $\mathbb{R}^p$ admet une unique partition en composantes connexes (respectivement par arc) maximales $X = \cup_i C_i$ telle que tout sur ensemble stricte de l'un des C_i n'est pas connexe (respectivement par arc).

Dans tout ce qui suit, nous nous intéressons à des ensembles finis de données. Pour un ensemble constitué de n points de $\mathbb{R}^p$, il est clair que l'ensemble des ses composantes connexes est formé des n singletons (un point par composante). Une telle décomposition a peu d'intérêt. Il faut donc étendre la notion de connexité au cas des ensembles finis, ce qui n'est possible que pour la connexité par arc. C'est pourquoi dans tout ce qui suit, nous ne traitons que de la connexité par arc, étendue comme dans la définition ci-dessous.

Définition 0.0.3 ($\delta-$connexité par arc): X est $\delta-$connexe par arc si et seulement si on peut lier tous les couples de points (x, y) de X par une suite de points de X (i.e. une suite de points de X) deux à deux distants d'au plus δ.

Par abus de langage, on confondra désormais $\delta-$connexité et $\delta-$connexité par arc.

Dans la première partie de cette thèse, nous proposons des méthodes de classification permettant de construire des classes connexes. Nous commençons par décrire les méthodes classiques (centres mobiles ou classifications hiérarchiques) pour observer qu'en général elles ne

conduisent pas à des classes connexes.

Nous montrons ensuite que la classification hiérarchique avec la distance du minimum permet d'obtenir toutes les partitions connexes (au sens des espaces discrets) possibles et nous proposons un indicateur de choix d'une partition adapté à la connexité. Cette méthode a de bonnes performances si les seuils de connexité (δ) sont suffisamment homogènes d'une classe à l'autre et si les classes sont assez éloignées les unes des autres. Nous étudions ensuite le problème de l'hétérogénéité des seuils de connexité δ.

Dans la deuxième partie, nous supposons construite une classe connexe, et nous définissons des méthodes de normalisation, de projection, de réduction de dimension et d'estimation de la dimension intrinsèque.

La troisième partie présente deux études appliquées à des questions économiques basées sur l'application des cartes de Kohonen à des données temporelles pour mettre en évidence des trajectoires individuelles (exemple des pays européens) et pour analyser des dynamiques de groupes (exemple des stratégies de gestion de portefeuille). Ainsi, il ne s'agit pas d'appliquer directement les méthodes des deux premières parties mais d'étudier l'algorithme de Kohonen dans le cas de classifications individuelles et temporelles. Ces études ont été réalisées à partir d'un programme convivial sous excel (en VBA) permettant de construire des cartes de Kohonen que j'ai réalisé pour le SAMOS.

Part I

CLASSIFICATION EN COMPOSANTES CONNEXES

INTRODUCTION

Dans toute cette partie, on s'intéresse à la séparation en composantes connexes d'un ensemble de points discrets.

On passe d'abord en revue les méthodes de classification les plus usuelles : classification des centres mobiles et classifications hiérarchiques. On constate qu'en général, elles ne permettent pas la mise en évidence de classes connexes.

Ensuite on présente rapidement un état de l'art, en mentionnant des méthodes alternatives à la nôtre, en décrivant dans chaque cas leurs avantages et leurs inconvénients.

Puis nous montrons que la classification hiérarchique avec la distance du minimum permet d'obtenir toutes les composantes connexes d'un ensemble. Mais il faut adapter la définition d'inertie intra-classes pour la rendre cohérente avec la notion de connexité.

Nous définissons alors le Minimum Spanning Tree (MST) et montrons que le comportement asymptotique des tailles des liaisons sur cet arbre (dans le cas où on utilise la distance du minimum) permet de savoir si une décomposition en plusieurs classes est pertinente (on teste l'existence d'au moins 2 classes connexes contre l'hypothèse de connexité globale).

Reste le cas difficile où les différentes composantes connexes ont des seuils de connexité hétérogènes. Pour résoudre cette difficulté, on construit dans le chapitre 5 de cette partie une méthode de classification reposant sur un couplage des problèmes de classification et d'estimation de densité qui donne de très bons résultats en estimation de densité (et segmentation) en dimension 1, mais pour laquelle l'extension en dimension quelconque n'est pas réaliste, compte tenu du trop grand nombre de points nécessaires. Dans le dernier chapitre de cette partie, on conjugue les deux techniques précédentes pour construire une méthodologie efficace de classification en composantes connexes.

1. LES MÉTHODES DE CLASSIFICATION NON SUPERVISÉES ET LA CONNEXITÉ

On dresse ici une liste non-exhaustive de méthodes de classification divisée en 3 parties :

- Des méthodes non compatibles, en général, avec l'obtention de classes connexes (sauf dans le cas connexe-convexe)
- Des méthodes construites en étroite relation avec la notion de connexité
- Des méthodes qui, bien que non construites en se fondant directement sur la notion de connexité, permettent de scinder un ensemble en composantes connexes.

1.1 Les méthodes classiques, qui en général ne conduisent pas à des classes connexes

Le but de ces méthodes classiques (méthode des centres mobiles, classifications hiérarchiques) est de construire des classes bien séparées et très homogènes. Elles reposent sur les notions d'inerties intra-classes et inter-classes, le but étant de minimiser l'inertie intra-classes en maximisant l'inertie inter-classes.

Rappelons les principales définitions :

Définition 1.1.1 (Inertie inter-classes, Inertie intra-classes): Notations :

- Soit $X = \{X_1, ..., X_N\} \subset \mathbb{R}^p$
- Soit G le barycentre de X
- Soit $cl : \{1, ..., N\} \to \{1, ..., K\}$ une classification des points, $cl(i)$ est le numéro de la classe du point X_i

- Soit G_k le barycentre de la classe de numéro k
- Soit n_k le nombre de points dans la classe de numéro k

On définit alors les inerties inter-classes et intra-classes par :

- Inertie inter-classes : $I_1 = \sum_{j=1}^{K}(n_j)||G_j - G||^2$
- Inertie intra-classes : $I_2 = \sum_{i=1}^{N}||X_i - G_{cl(i)}||^2$

On rappelle que I l'inertie totale définie par $I = \sum_{i=1}^{N}||X_i - G||^2$ est égale à $I_1 + I_2$ (quelque soit la classification). Donc, pour un nombre de classes fixé K, la minimisation de l'inertie intra-classes est équivalente à la maximisation de l'inertie inter-classes.

On voit tout de suite que la notion de barycentre n'est pas bien adaptée aux classes connexes. On voit par exemple dans la figure 0.1 de l'introduction que les deux classes du premier exemple ont le même barycentre alors que pour le deuxième exemple le barycentre d'une classe appartient à l'autre (et réciproquement).

1.1.1 Classification par la méthode des centres mobiles

La méthode des centres mobiles est une méthode de classification mise au point de manière simultanée et sous plusieurs noms ($K - means$, nuées dynamiques...) par, notamment, Lloyd ([KM3] 1982), Forgy ([KM1] 1965) ou Macqueen ([KM4] 1967).

Dans cette méthode, le nombre de classes est choisi à l'avance. L'algorithme est très simple. On se donne initialement K centres G_k arbitraires dans l'espace $\mathbb{R}^p$. L'algorithme est itératif, à chaque étape :

- On affecte chaque point X_{i_0} à la classe k telle que $k = arg\min_i\{d(X_{i_0}, G_i)\}$
- On adapte les centres : pour chaque k de 1 à K, G_k est recalculé comme le barycentre de tous les points affectés à la classe k

On démontre que cette méthode minimise l'inertie intra-classe et qu'elle converge vers un minimum ne dépendant que des valeurs initiales des centres. Par définition de l'algorithme, les classes sont séparées par des hyperplans (hyperplans médiateurs des segments joignant deux

barycentres) et cet algorithme ne conduit pas à des classes connexes en général.

On observe sur la figure 1.1 le résultat de l'algorithme des centres mobiles appliqué au problème des deux cercles concentriques (on a pris $K = 2$ et G_1, G_2 tirés aléatoirement sur $[0, 1]^2$. On voit clairement que l'algorithme ne retrouve pas les classes connexes.

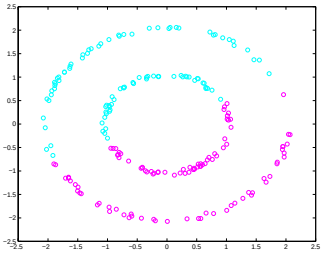


Fig. 1.1: Résultat d'un $K - means$ en 2 classes sur des cercles concentriques

1.1.2 Classifications hiérarchiques

On rappelle que la distance d entre les points est la distance euclidienne usuelle. Une classification hiérarchique consiste à construire une suite de classifications emboîtées des N points en $N, N-1, N-2, \ldots, 1$ classes. A l'état initial on considère la partition en N singletons $\{\{X_1\}, ..., \{X_N\}\}$ et à chaque étape, on passe d'une partition en K classes $\{C_1^K, .., C_K^K\}$ à une partition en $K-1$ classes en agrégeant les deux classes C_i^K et C_j^K les plus proches. Il faut donc définir une distance entre ensembles qu'on notera D. Différents choix sont possibles.

Les principales distances utilisées sont les suivantes ([HIE2] 1973, [HIE6] 1971, [HIE9] 1974, [HIE17] 1963):

Si A et B sont deux sous-ensembles de X d'effectifs N_A et N_B, de barycentre G_A et G_B :

- **distance du minimum :** $D(A, B) = min\{d(a, b), a \in A, b \in B\}$
- **distance du maximum :** $D(A, B) = max\{d(a, b), a \in A, b \in B\}$
- **distance de la moyenne :** $D(A, B) = \frac{1}{N_A N_B} \sum d(a, b)$
- **distance des barycentres:** $D(A, B) = d(G_A, G_B)$
- **distance de Ward:** $D(A, B) = \frac{N_A N_B}{N_A + N_B} d(G_A - G_B)$

Comme on le verra par la suite (section 3) parmi toutes les distances possibles, seule la distance du minimum permet d'obtenir des classes connexes.

Dans la liste des distances entre classes usuelles, c'est la distance de Ward qui est la plus fréquemment utilisée, car elle minimise l'accroissement de l'inertie intra-classes à chaque étape. Elle permet d'obtenir pour un même nombre de classes une inertie intra-classes plus petite.

Si on applique la classification hiérarchique avec distance de Ward au cas des deux cercles concentriques, on constate là aussi que la segmentation obtenue ne met pas en évidence les classes connexes (voir figure 1.2).

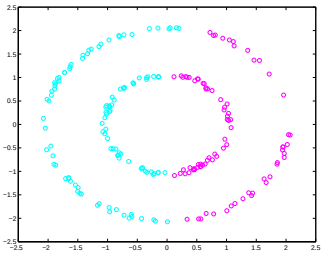


Fig. 1.2: Résultat d'une classification hiérarchique par la méthode de Ward en 2 classes sur des cercles concentriques (même données que pour l'illustration des $K-means$)

1.2 Les méthodes reposant sur la connexité

1.2.1 Classification et traitement d'image

C'est en traitement d'image qu'on voit le plus souvent citées conjointement classification et connexité puisqu'on cherche à segmenter les images suivant les caractéristiques (couleur, niveaux de gris...) des pixels sur des composantes connexes des pixels (composantes connexes au niveau spatial i.e. suivant leur localisation géographique) ([PAT1], [PAT2], [PAT3]). L'image sur laquelle on travaille constitue une composante géographique "complète", dans le sens où à chaque coordonnée de l'image, discrétisée, correspond un pixel. En théorie, on peut donc facilement caractériser un ensemble de pixels connexes. C'est ce dernier point qui rend les méthodes utilisées en image difficilement adaptables au cadre général de l'analyse des données.

1.2.2 Les méthodes reposant sur l'estimation de densité

Les méthodes fondées sur des estimations de densité, principalement la méthode du water-shed ([DST1]) et la méthode de regroupement dans les domaines d'attraction des modes de Wishart ([DST7]) reposent directement sur la notion de connexité.

La méthode du "water-shed"

En dimension 2 la méthode du "water-shed" est très imagée : si on assimile les données à des coordonnées géographiques et la densité à leur altitude, on obtient un "paysage" que l'on immerge dans de l'eau et on observe les "îles" qui apparaissent. Les points sont alors classés en fonction de l'île sur laquelle ils se trouvent éventuellement. De manière plus mathématique, et en dimension quelconque, on se fixe un seuil λ ("niveau d'eau") et on classe les points en fonction de l'appartenance à l'une ou l'autre des composantes connexes de $E_\lambda = \{x/f(x) > \lambda\}$, où f est la densité ayant sous-tendu le tirage. La définition même de la méthode met en évidence l'étroite liaison du "water-shed" et de la connexité.

Une telle méthode montre très vite ses limites. Il y a d'une part le choix arbitraire du paramètre λ et, d'autre part, dans des cas d'hétérogénéité comme dans l'exemple de la figure 1.3, l'incapacité théorique à trouver conjointement toutes les "vraies classes".

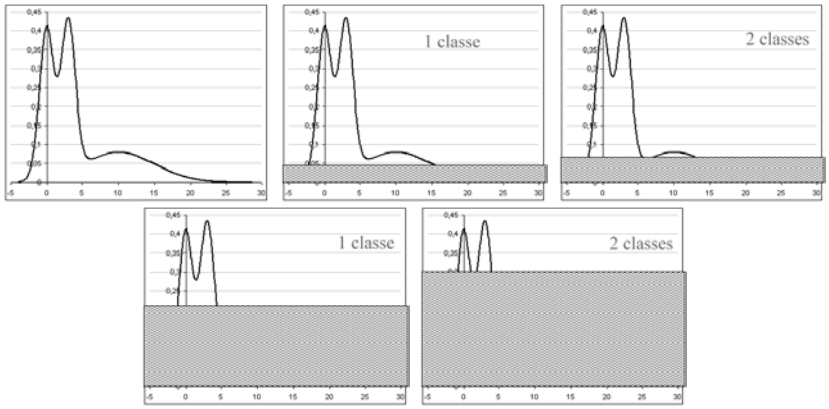


Fig. 1.3: Méthode du "water-shed"

Attraction des modes

Pour répondre aux problèmes posés par la méthode du "water-shed", Wishart ([DST7]) a proposé une nouvelle méthode en 1969 : la classification autour des domaines d'attraction des modes. Dans cette méthode il y a autant de classes que de modes de la densité et chaque point est affecté à la classe d'un mode si on peut relier le point au mode par un chemin continu le long duquel la densité est croissante.

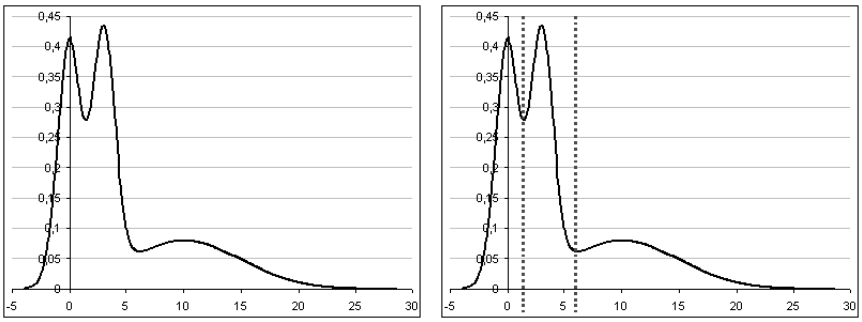


Fig. 1.4: Méthode du domaine d'attraction des modes

On peut présenter les résultats sous la forme d'un dendrograme dont les feuilles correspondent aux modes et les regroupements entre classes aux "points" selles de la densité (en dimension 1 les minima locaux, en dimension supérieur les points selles).

Pour comprendre le lien entre cette méthode et la connexité, il faut tout d'abord observer que cette méthode constitue une "amélioration" de la méthode du "water-shed" elle-même étroitement liée à cette notion. On soulignera aussi que, si on se place dans le cas "continu", le fait de regrouper les points dans les domaines d'attraction des modes impose la connexité par arc des classes (chaque point étant affecté à la classe d'un mode si il existe un chemin continu le menant au mode avec une densité croissante sur le chemin, chaque couple de points d'une classe est lié par un chemin continu).

Dans le 4^{eme} chapitre de la partie classification nous affinerons ces méthodes. Nous ne les appliquerons pas après avoir estimé la densité, mais en effectuant classification et estimation de densité conjointement. En effet, après avoir montré ici en quoi la connaissance de la densité aide à classer les données, on montrera que la connaissance de la classification est un atout en estimation de densité et on construira un algorithme reposant sur le couplage des deux méthodes.

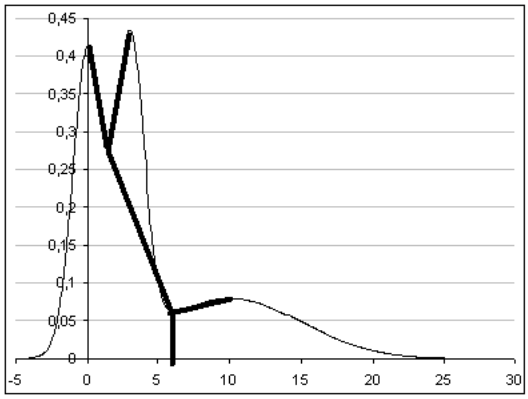


Fig. 1.5: Méthode du domaine d'attraction des modes : dendogramme associé

1.2.3 DBSCAN

La méthode DBSCAN ([DST8]), initiée par Ester et Al en 1996 repose sur la notion de connexité par arc. Pour savoir si deux points sont liés entre eux on se fixe un seuil δ et on définit le $\delta-$voisinage d'un point X comme l'ensemble des observations distantes de moins de δ de X.
On définit alors l'ensemble des points "directement" atteignables de X, en se donnant un nouveau paramètre n et en définissant alors :

Y est directement atteignable par X si et seulement si Y est dans un $\delta-$voisinage de X et le δ-voisinage de X contient au moins n points.

Une telle définition tend a éliminer les "problèmes de bords".

Enfin on dira que Y est lié a X si il existe un chemin (suite d'observation) $X_1, ..., X_k$ tel que $X_1 = X$, $X_k = Y$ et, pour tout i X_{i+1} est directement atteignable par X_i.

Pour finir X et Y sont "liés par densité" si X est lié a Y ou Y est lié a X.

Enfin, X est dans la même classe que Y si les deux points sont liés.

DBSCAN donne de bons résultats, bien sur le choix des paramètres est, comme toujours délicat mais plusieurs indicateurs sont proposés. Le plus gros inconvénient d'une telle méthode est d'assimiler les classes les plus fortement dispersées à du "bruit". Cette méthode sera donc pertinente dans le cas où l'on cherche a "éliminer" un bruit sous-jacent mais ne permettra jamais de retrouver deux classes fortement hétérogènes en dispersion.

1.3 Des méthodes permettant de trouver des classes connexes

1.3.1 Distance de marche aléatoire

Cette méthode repose sur la construction d'un graphe G pondéré par la longueur des liaisons entre les données : $G_{i,j} = ||X_i - X_j||$ si les points X_i et X_j sont connectés et 0 sinon.

On utilise une fonction f de $\mathbb{R}^+$ dans $\mathbb{R}^+$ décroissante pour définir des probabilités $p_{i,j}$:

$$p_{i,j} = f(G_{i,j}) / \sum_{k \ G_{i,k}>0} f(G_{i,k})$$

Le principe est alors le suivant : la distance entre deux points est le "temps de parcours moyen" (calculé comme un nombre d'étapes) entre les points lorsqu'on considère une marche aléatoire avec une probabilité de passer de l'état i à l'état j qui vaut $p_{i,j}$

Le fait d'avoir choisi f décroissante est équivalent à dire que la probabilité de passer d'un point à un des points qui lui sont connectés par G est d'autant plus élevée que leur distance est faible.

Plusieurs type de graphes G et fonction f sont fréquemment utilisés dans la littérature, le couple de graphe et fonction le plus souvent rencontré est le suivant :

$f(x) = exp(-x^2/\sigma^2)$ avec un paramètre σ choisi par l'utilisateur.

Soit $m(j,i)$ le temps moyen (nombre moyen d'étapes) qu'il faut pour atteindre X_j (une première fois) en partant de X_i pour une marche aléatoire dont la probabilité de transition pour passer d'un point X_k au point X_l est $p_{k,l}$.

De manière recursive on a :

$$m(i,i) = 0$$

$$m(j,i) = 1 + \sum_{k,k\neq j} p_{i,k} m(k,j)$$

(On regarde le nombre d'étapes nécessaires pour atteindre j par rapport au nombre d'étapes nécessaires pour atteindre n'importe quel autre point).

Comme $m(i,j)$ n'est pas symétrique, pour définir une distance, on prendra : $n(i,j) = m(i,j) + m(j,i)$ temps moyen pour faire l'aller retour de X_i à X_i en passant par X_j.

Il a été montré que la distance $n(i,j)$ peut être calculée rapidement par inversion d'un système linéaire, ce qui donne aussi aux méthodes de classification par distance de marches aléatoires le nom de classification spectrale.

Une fois la distance entre points définie, toutes (ou presque) les méthodes de classifications classiques peuvent être adaptées avec cette nouvelle distance (on peut citer, par exemple l'utilisation de la classification hiérarchique dans [SPC6] et des $K-means$ dans [SPC3] et [SPC5])

Les résultats en terme de classification non supervisées avec des composantes connexes sont excellents :

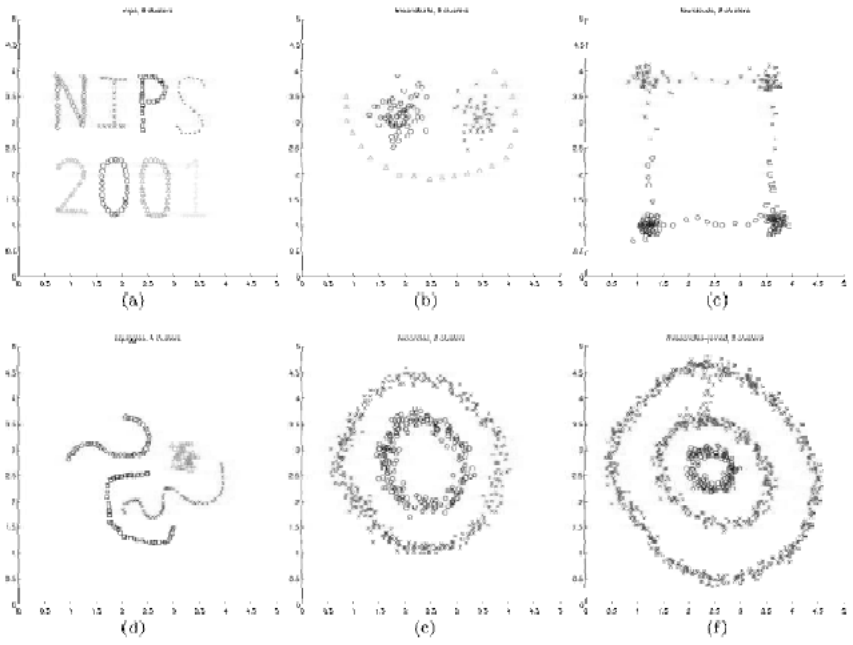


Fig. 1.6: Résultats obtenus par Ng, Jordan et Weiss dans "On Spectral Clustering : Analysis and an algorithm"

Si la notion de connexité n'est pas explicitée dans la démarche de la classification par distance de marches aléatoires, il est clair qu'elle est sous-jacente à la problématique. En effet la distance de marche aléatoire est évidemment liée à la notion de connexité par arc : on observe les

possibilités de lier les points sur des chemins relativement courts (pas trop d'étapes sur les liaisons) dont les pas sont eux-mêmes courts (sinon ils ont une probabilité faible d'arriver).

Les méthodes de classifications par des distances de marche aléatoire donneront des résultats plutôt meilleurs que les nôtres (pour la méthodologie finale) mais nécessitent le choix d'un certain nombre de paramètres : type de graphe, fonction et paramètres de la fonction, nombre de classes... alors que la dernière méthode de classification que nous allons proposer est automatique (ou tout du moins, à chaque fois que le choix d'un paramètre sera nécessaire, des indicateurs seront disponibles). Une voie à explorer serait de paramétrer les méthodes par distance de marche aléatoire à l'aide des résultats des méthodes proposées dans cette thèse, pour gagner en robustesse.

1.3.2 Les "Growing Neural Gas"

Mis au point par Fritzke en 1994 dans ([GNG2]), les Growing Neural Gas (GNG) ne constituent pas, à proprement dit une méthode de classification. Dérivée des cartes de Kohonen, cette méthode place des neurones et les relie de manière à respecter une topologie qui se définit au fur et à mesure.

L'algorithme se résume ainsi :

A chaque étape les centres s'adaptent aux données : on va tirer aléatoirement un point dans l'ensemble des observations et déplacer les deux centres les plus proches de ce point vers lui et créer une liaison entre ces deux centres. Les liaisons disparaissent en fonction de leur "âge" (calculé en nombre d'itérations d'existence). De nouveaux centres apparaissent avec une fréquence fixée (en nombre d'itérations). Le déplacement des centres et le fait que la topologie, caractérisée par les liaisons entre centres, s'adapte au cours de l'algorithme font de GNG (ou Growing Neural Gaz) un algorithme très performant.

Pour une meilleure compréhension de l'algorithme lui même, en voici une version plus détaillée :

- Initialisation des neurones ou centres "mobiles": on choisit les deux premiers neurones au hasard dans l'ensemble de points. On obtient $A = \{c_1, c_2\}$, A étant dans toute la suite l'ensemble des neurones (confondus avec leur représentants)

- Initialisation des connexions entre neurones $C \subset A \times A$: à l'initialisation $C = \emptyset$
- On itère alors
  - Tirage aléatoire d'un point X_i dans la base
  - Recherche dans A de s_1 et s_2 tels que :

$$s_1 = arg \min_{s \in A} ||X_i - s||$$

$$s_2 = arg \min_{s \in A|_{\{s_1\}}} ||X_i - s||$$

    qui sont les premier et deuxième voisins de X_i dans l'ensemble des neurones
  - Si la connexion entre s_1 et s_2 n'existe pas, on la crée ($C := C \cup \{\{s_1, s_2\}\}$) et on met son âge à 0
  - on incrémente l'erreur locale en s_1 : $E_{s_1} := E_{s_1} + ||X_i - s_1||^2$
  - On déplace les neurones s_1 et s_2 vers X_i :

$$s_1 := (1 - \varepsilon_1)s_1 + \varepsilon_1 X_i$$

$$s_2 := (1 - \varepsilon_2)s_1 + \varepsilon_2 X_i$$

  - On augmente tous les âges des connexions qui partent de s_1 de 1
  - On supprime toutes les liaisons d'âges supérieurs à age_{max}
    - Si l'itération est un multiple d'une période T fixée, on ajoute un neurone de la manière suivante :
    - On recherche le neurone pour lequel il y a le plus d'erreur accumulée $s = argmax\{E_s\}$
    - Parmi les neurones connectés à s, on recherche le neurone ayant accumulé le plus d'erreur $t = argmax\{E_t, (t, s) \in C\}$
    - On ajoute un nouveau neurone q qui est le barycentre de t et de s, et on le lie à t à s. On supprime la liaison (t, s) de C
    - On actualise les erreurs accumulées

A la suite de l'algorithme, on peut construire de manière naturelle une classification des données : tout d'abord on effectue une classification sur les neurones en fonction des composantes connexes du graphe des liaisons, puis on affecte chaque point à la classe de son plus proche neurone.

Cette méthode sera relativement efficace si on dispose de beaucoup de données et si les classes sont suffisamment séparées. Si les classes sont trop proches, des liaisons entre les classes apparaîtront ponctuellement et il faudra, par exemple, seuiller leur fréquence d'apparition pour ne conserver que les liaisons "stables".

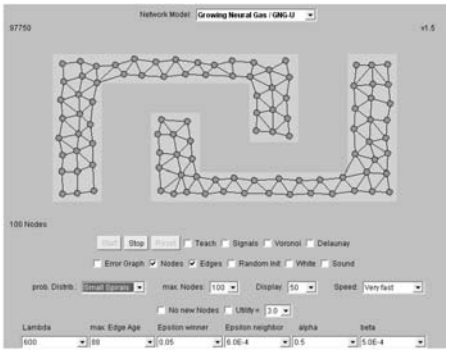


Fig. 1.7: Capture d'écran sur le site de Fritzke ([GNG4]) : un cas où la classification à l'aide de GNG fonctionne

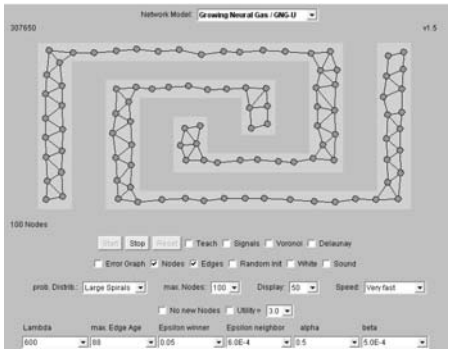


Fig. 1.8: Rapprochement des classes : les liaisons stables

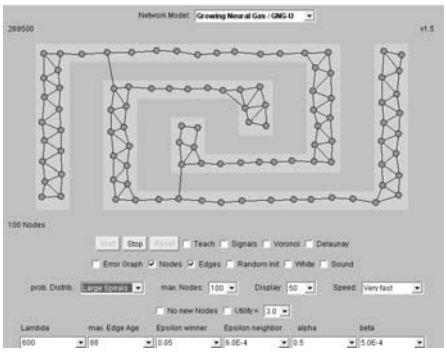


Fig. 1.9: Rapprochement des classes : apparition d'une liaison ponctuelle

1.3.3 Les Cartes de Kohonen

Comme les GNG, les cartes de Kohonen ([SOM2]) ou Self Organizing Maps, ne sont pas à l'origine, un algorithme de classification, mais un algorithme de projection non linéaire des données sur un espace de dimension réduite (le plus fréquemment 1 ou 2) en respectant le mieux possible la topologie des données. La lecture aisée d'une carte de Kohonen a permis de mettre au point des méthodes simples de visualisation des classes (la $U-$matrice ([SOM10]) la $P-$matrice ([SOM9]) ainsi que plusieurs autres dérivées).

L'algorithme des cartes de Kohonen est le suivant :

On se fixe une structure (le plus souvent une "ficelle" ou une "carte" mais on peut aussi choisir des structures moins élémentaires telle que des tores, des cylindres...). Par structure, on entend un ensemble de cellules et une fonction de voisinage V définie sur cet ensemble.

Par exemple :

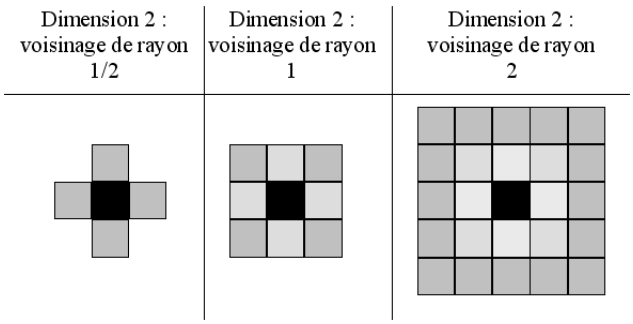


Fig. 1.10: Exemples de voisinages pour une grille à cellule "carrées"

Chaque cellule est représentée par un vecteur code dans l'espace des observations (à chaque case (i, j) correspond un vecteur code $C_{i,j}$).

On se donne alors deux fonctions : $r(t)$ et $\varepsilon(t)$ respectivement "rayon" et "gain" ainsi qu'un nombre d'itérations N_{it} et on itère N_{it} fois:

- Tirage aléatoire d'un individu i_0 dans la base
- Recherche du vecteur code $C_{i,j} = argmin_{i,j}\{||C_{i,j} - X_{i0}||\}$ le plus proche du point tiré
- Pour tout (i', j') tel que $V((i, j), (i', j')) \leq r(t)$ on effectue $C_{i',j'} := (1 - \varepsilon(t))C_{i',j'} + \varepsilon(t)X_{i0}$

On construit ainsi une projection des données sur une structure de plus petite dimension, caractérisée par les vecteurs codes et la topologie. Dans le cas de la dimension 1 ou 2, la représentation graphique est aisée et à partir de cette "carte" on peut classer et visualiser les données. Plusieurs méthodes existent pour cela :

- Classification sur les vecteurs codes, le plus souvent une classification hiérarchique avec la distance de Ward
- La $U-$Matrice ([SOM10]), matrice des distances entre chaque vecteur codes et la moyenne des vecteurs code qui l'entourent. Une grande valeur indiquera une rupture de continuité dans la carte et en "coloriant" les cases avec un niveau de gris proportionnel à cette valeur, les classes se lisent comme les parties séparées par les frontières les plus sombres
- La méthode de la $P-$Matrice ([SOM9]) repose sur des estimations de densité. On colorie les cases de la carte avec des niveaux de gris proportionnels à la valeur d'une estimation de densité, alors les classes seront séparées par des frontières "claires"
- Plusieurs méthodes combinant les deux types de matrices précédentes existent aussi, par exemple ([SOM13])
- On peut aussi construire un indicateur graphique permettant de lire la distance entre chaque "cellule" et ses voisines à l'aide de polygones ([SOM18])

De telles méthodes fonctionnent bien pour des données qui se projettent correctement en dimension 2.

Bien sûr on peut élargir l'algorithme de Kohonen à des topologies de dimension supérieure à 2 (ce qu'on ne se privera pas de faire dans la partie suivante) mais alors la représentation des données et la lecture graphique des classes devient impossible.

Le lien entre les classifications par cartes de Kohonen et la connexité vient du respect de la topologie lors de la projection.

2. CLASSIFICATION HIÉRARCHIQUE AVEC LA DISTANCE DU MINIMUM

Dans ce chapitre on montre que la classification hiérarchique par la distance du minimum, qui est certainement une des plus ancienne méthodes de classification (cf Sneath [HIE18] en 1957 et Johnson [HIE13] en 1967) mène à l'obtention de toutes les partitions en composantes $\delta-$connexes maximales (voir l'introduction pour les définitions). Comme on l'a vu en introduction, l'inertie intra-classe usuelle ne sera pas un bon indicateur pour le choix du nombre de classes. Nous définirons alors une nouvelle inertie intra-classe, ne reposant pas sur la distance euclidienne, mais sur une distance liée à la connexité qui nous permettra, par lecture graphique, de choisir un nombre de classes "correct" à l'issue de la classification. Les différentes propriétés, résultats et notions présentés dans ce chapitre ne sont pas nouveaux, mais on s'attachera surtout à montrer l'étroit lien entre cette classification hiérarchique et la connexité. En effet, par la suite on utilisera fréquemment comme outil le Minimal Spanning Tree (MST) étroitement lié à la classification hiérarchique avec la distance du minimum (pour le lien entre MST et classification voir Gower et Ross 1969 [MST11]).

2.1 Classification hiérarchique et $\delta-$connexité

Proposition 2.1.1: Soit E un ensemble fini de cardinal n. Alors pour un δ fixé, il existe une unique partition $\delta-$connexe maximale (au sens de l'inclusion, et minimale au sens du nombre de composantes connexes).

Démonstration :

Unicité : Supposons qu'il existe deux partitions différentes et $\delta-$connexes maximales en p classes : $\{F_1, ..., F_p\}$ et $\{G_1, ..., G_p\}$. Alors il existe un point x dans deux classes différentes (quitte à ré-indicer, $x \in F_1$ et $x \in G_1$ avec, par exemple $G_1 \subsetneq F_1$). Il existe alors un point y qui appartient a G_1 mais pas à F_1. Soit alors F_i la classe de y, quitte

a ré-indicer, on suppose que $y \in F_2$, on a alors : $(F_1 \cup F_2, F_3, ..., F_p)$ qui réalise une partition connexe de $p-1$ classes. Ce qui contredit la maximalité de la partition.

Existence : La fonction qui, à une partition, associe son nombre de classes est à valeurs dans $\{1, ..., n\}$, donc admet un minimum et l'atteint.□

On peut alors définir $p(\delta)$ fonction qui à δ associe le nombre de classes de la partition minimale $\delta-$connexe. Cette fonction est décroissante et en escalier. On notera :
$\delta_{max}(p) = sup\{\delta, p(\delta) = p\}$
$\delta_{min}(p) = inf\{\delta, p(\delta) = p\}$
avec : $\delta_{max}(p) = \delta_{min}(p-1)$

Proposition 2.1.2: l'algorithme de classification hiérarchique par la distance minimum $(d(A,B) = min\{d(a,b), a \in A, b \in B\}$ mène à l'obtention des n segmentations $\delta-$connexes minimales possibles

Démonstration : Le passage de la classification $\delta-$connexe minimale en p classes $(C_1^p, ..., C_p^p)$ à la classification $\delta-$connexe minimale en $p-1$ classes $(C_1^{p-1}, ..., C_{p-1}^{p-1})$ se fait par agglomération des deux classes les plus proches au sens de la distance min.

Soit $m = min(d(C_i^p, C_j^p))$ et, quitte à ré-indicer $(1,2) = argmin(d(C_i^p, C_j^p))$. La partition $(C_1^p \cup C_2^p, C_3^p, ..., C_p^p)$ est $m-$connexe. Elle est de plus $m-$connexe minimale car :

$\forall \delta < m$ $(C_1^p, ..., C_p^p)$ est $\delta-$connexe
et $\forall \delta > m$ $(C_1^p, ..., C_p^p)$ n'est pas $\delta-$connexe □

2.2 Distance intra-classes

On cherche maintenant à construire une notion de distance intra-classes compatible avec la notion de connexité afin de déterminer un indicateur du nombre de classes "optimal" à choisir.

2.2.1 Distance entre deux points

Comme on l'a dit en introduction, la notion d'inertie intra-classes classique ne va pas être un bon indicateur dans le cas de la connexité. Ceci provient du fait que si deux points sont proches (pour la distance euclidienne) ceci n'implique rien sur leur "seuil de liaison". La figure suivante illustre cette affirmation. Dans les deux exemples les points A et B de la figure sont situés à la même distance euclidienne mais il est "visible" qu'une distance tenant compte de la connexité doit être plus élevée dans le premier cas que dans le second. :

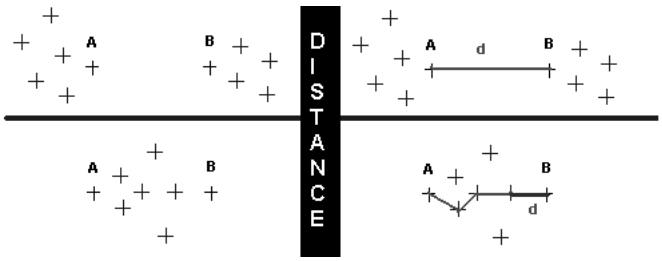


Fig. 2.1: Construction d'une distance compatible avec la notion de connexité

Pour résoudre ce problème on définit la distance entre deux points de E comme le plus petit des plus grands sauts effectués lorsqu'on lie x_i à x_j par un chemin de points de E.

Un chemin che de longueur n de points de E liant x_i à x_j est une application de $\{1, ..., n\}$ dans $\{1, ..., N\}$ telle que $che(1) = i$ et $che(n) = j$.

Sur un chemin donné che, le saut maximum est :

$$saut_{max}(x_i, x_j, che) = max\{d(x_{che(i)}, x_{che(i+1)})\}$$

Ce saut représente la plus grande distance (euclidienne) entre deux points du chemin. On note $CHE(x_i, x_j)$ l'ensemble des chemins de longueur quelconque de points de E liant x_i à x_j. La nouvelle distance associée à la distance initiale d est définie par :

$$d_c(x, y) = min\{saut_{max}(x_i; x_j, che)\}.$$

C'est-à-dire dire qu'on cherchera la plus petite valeur (sur l'ensemble des chemins) du plus grand saut effectué sur chaque chemin.

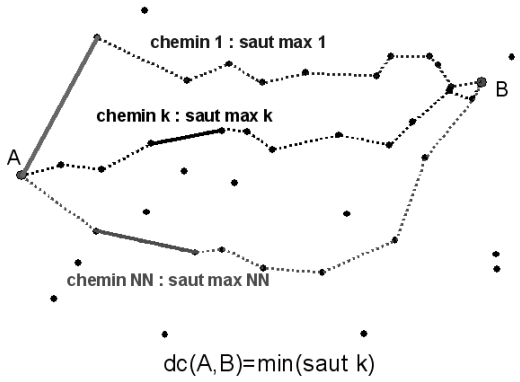


Fig. 2.2: Construction d'une distance compatible avec la notion de connexité : le plus petit saut maximal

On vérifie qu'on a bien défini une distance dans un ensemble fini de points deux à deux distincts :

- $d_c(x, y) = 0 \Rightarrow x = y$ dans un ensemble discret (ce qui serait faux autrement).
- $d_c(y, x) = d_c(x, y)$ de manière évidente en inversant l'ordre des chemins.
- $d_c(x, z) \leq max(d_c(x, y), d_c(y, z))$ car dans la seconde partie de l'inégalité on contraint les chemins à passer par le point y. De cette inégalité découle l'inégalité triangulaire : $d_c(x, z) \leq d_c(x, y) + d_c(y, z)$

Caractérisation :

Pour un couple de points donnés le nombre de chemins les liant est : $\sum_{k=1}^{n-2} A_n^k$ (avec A_n^k le nombre de $k-$arrangements qui vaut $n!/(n-k)!$). Il est alors évident que le calcul de tous les chemins possibles est excessivement coûteux en temps et rend l'obtention de la distance non réaliste sans autre caractérisation. Heureusement on montre ici qu'à l'issue de la classification hiérarchique, on obtient de manière immédiate la distance d_c entre tous les couples de points par la formule suivante :

Théorème 2.2.1: Si on note $C_k(x)$ la classe du point x dans une classification en k classes :

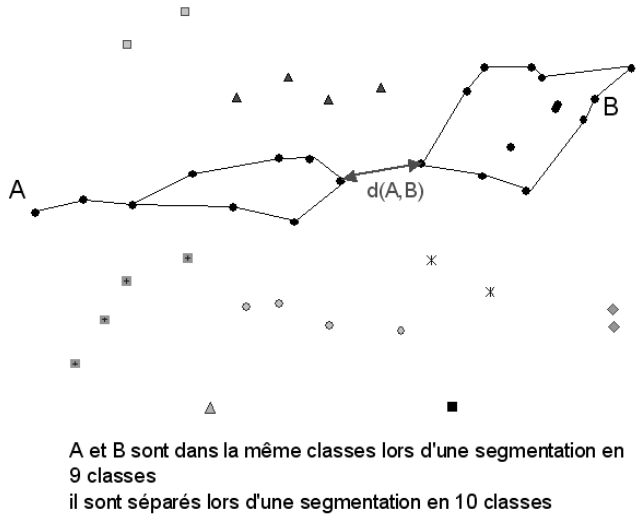


Fig. 2.3: Caractérisation de la distance (même ensemble que précédemment) le plus petit saut maximal correspond à la distance (minimum) entre les "plus grandes" classes qui les séparent

$d_c(x,y) = \delta_{min}(max\{k/C_k(x) = C_k(y)\})$

ou de manière équivalente :

$d_c(x,y) = \delta_{max}(min\{k/C_k(x) \neq C_k(y)\})$

Démonstration :

On a égalité entre :
$\delta_{min}(max\{k/C_k(x) = C_k(y)\})$ et $\delta_{max}(min\{k/C_k(x) \neq C_k(y)\})$

car $max\{k/C_k(x) = C_k(y)\} = min\{k/C_k(x) \neq C_k(y)\} + 1$

et $\delta_{min}(p+1) = \delta_{max}(p)$.

De plus : de manière évidente $d_c(x,y) \leq \delta_{min}(max\{k/C_k(x) = C_k(y)\})$ car :

$\forall \delta \geq \delta_{min}(max\{k/C_k(x) = C_k(y)\})$, x et y sont dans la même classe $\delta-$connexe donc reliable par un chemin de points deux à deux distants

d'au plus δ. Ainsi $d_c \leq \delta$.

De même $d_c(x, y) \geq \delta_{max}(min\{k/C_k(x) = C_k(y)\})$ car :

$\forall \delta \leq \delta_{max}(min\{k/C_k(x) \neq C_k(y)\})$ x et y ne sont pas dans la même classe $\delta-$connexe. Donc on ne peut pas les relier par un chemin de points deux à deux distants d'au plus δ donc on a $d_c \geq \delta$.□

On remarque que, dans un ensemble fini de cardinal n, l'ensemble des distances entre couples de points prend ses valeurs dans un ensemble de n nombres.

On a définit cette distance pour montrer, encore une fois, l'intérêt de la classification hiérarchique avec la distance du minimum pour la connexité, cette distance vérifie d'autres propriétés non-mentionnées ici (tel, par exemple, le fait que ce soit l'ultramétrique sous-dominante [HIE14]).

2.2.2 Distance intra-classes

Nous avons ainsi calculé la distance intra-classes comme une moyenne des distances d_c entre couples de points au sein de chaque classe. En appelant $D_c(p)$ la distance intra classe d'une classification en p classes maximale on a :

$$D_c(p) = \sum_{i,j} \mathbf{1}_{\{\mathbf{C_p(x_i)=C_p(x_j)}\}} \mathbf{d_c(x_i, x_j)}$$

La caractérisation précédente des distances entre couples de points à l'aide des sorties de la classification hiérarchique montre que l'algorithme de classification hiérarchique avec la distance min est l'algorithme qui permet de minimiser la distance intra-classes D_c sous contrainte d'un nombre de classes fixé.

On préférera néanmoins utiliser :

$$\overline{D_c(p)} = (1/N_p) \sum_{i,j} \mathbf{1}_{\{\mathbf{C_p(x_i)=C_p(x_j)}\}} \mathbf{d_c(x_i, x_j)}$$

avec : $N_p = \sum_{i,j} \mathbf{1}_{\{\mathbf{C_p(x_i)=C_p(x_j)}\}}$
C'est la distance intra-classes "moyenne"

2.2.3 Choix du nombre de classes

De manière classique, on va choisir la classification provoquant une plus grande "homogénéisation" des données. Ce nombre de classes sera caractérisé par une forte rupture de distance intra-classes qui sera synonyme de fort changement de structure vers une plus forte similitude des données. On observera donc le diagramme des distances intra-classes pour tenter d'y déterminer un saut important.

On notera dans la suite $Saut_{Dintra}(p) = (\overline{D_c(p)} - \overline{D_c(p-1)})$ le saut relatif de distance intra-classes lorsqu'on passe d'une classification en $p-1$ classes à une classification en p classes.

2.3 Quelques résultats sur des exemples

2.3.1 Séparation de gaussiennes

Dans cet exemple on observe le résultat de la classification appliquée à la séparation de réalisations de deux gaussiennes normées en dimension deux (de variance identité I_2), l'une centrée, l'autre de moyenne (m, m). 100 points ont été tirés dans chacune des deux classes. Lorsque les gaussiennes sont suffisamment éloignées, on les sépare parfaitement (premier exemple), puis à mesure qu'on les rapproche il faut isoler certains points pour retrouver les deux composantes connexes principales.

2.3.2 Exemples classiques de classes convexes

Il s'agit de deux exemples classiques en classification : la base de Ruspini et les Iris de Fischer. Dans ces deux exemples, les classes sont connexes et convexes mais considérées comme difficiles a séparer. Dans les deux cas on retrouve de bonnes classifications (dans le sens ou on sait ici quelles sont les vraies classes") non pas pour le plus grand saut de distance intra-classes mais pour le dernier saut significatif. Dans le cas des Iris, on est obligé d'isoler un grand nombre de points avant de trouver trois classes centrales.

2.3.3 Exemples classiques de classes connexes et non convexes

Nous traitons ici des exemples de classes connexes et non convexes. Contrairement aux cas précédents, les méthodes de classification non

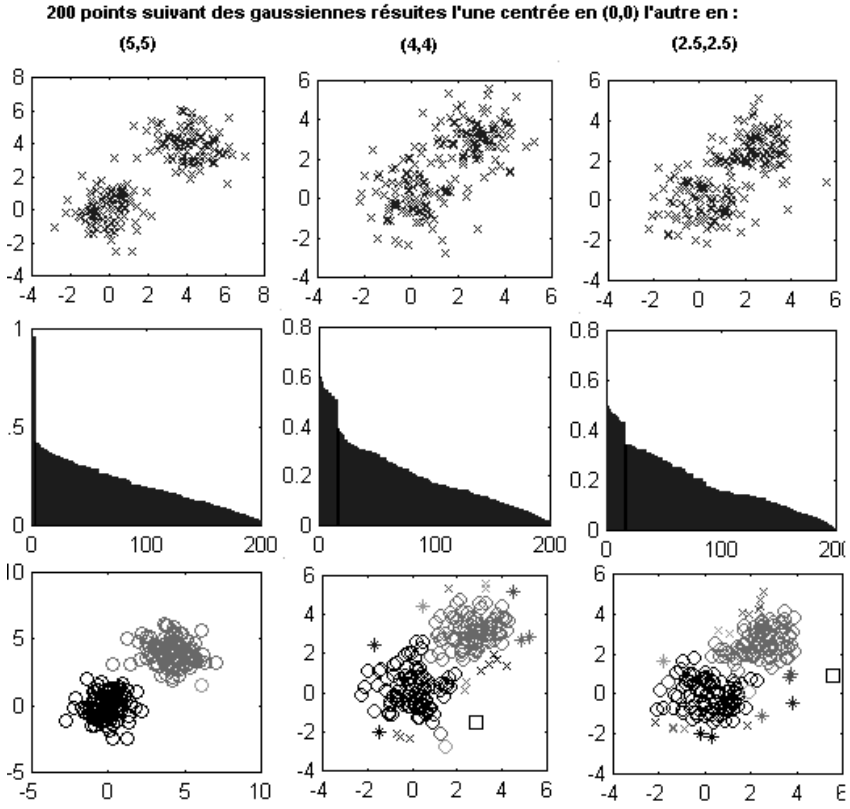
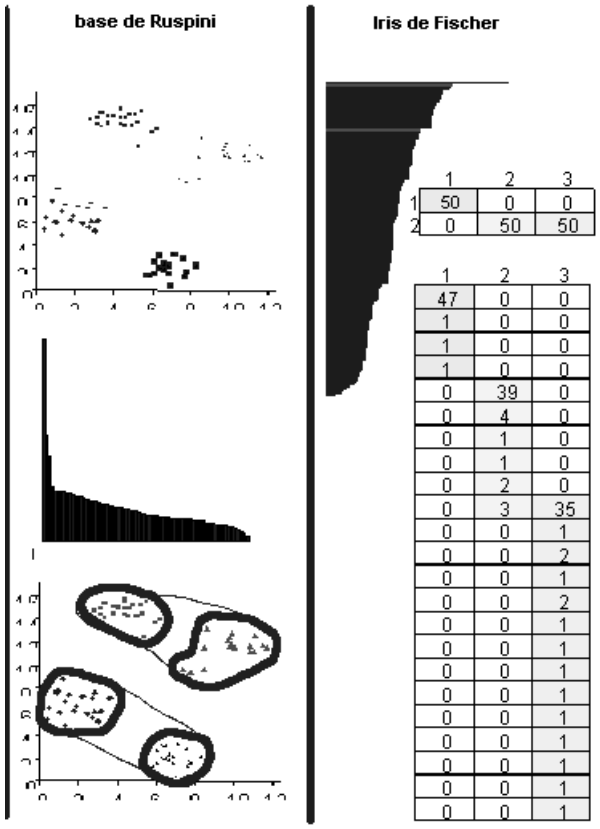


Fig. 2.4: Séparation de deux gaussiennes en dimension 2 : verticalement nuage de point, distance intra-classes pour chaque partition de la hiérarchie et classification associée à la plus grande rupture de distance intra-classes

supervisées "classiques" reposant sur des séparations de l'espace par des hyper-plans ou des regroupements autour de barycentres ne sont pas efficaces du fait de la forme même des classes.

On voit que, tant qu'il existe une certaine homogénéité entre les dispersions au sein des classes relativement à l'éloignement entre les classes (i.e. dans tous les cas sauf le dernier), quitte à isoler certains points éloignés de tous les autres, on retrouve bien les deux classes principales. Les graphiques étant en noir et blanc, on a noirci les composantes connexes "à la main" pour visualiser les classes. Dans les deux premiers cas, on retrouve parfaitement les deux cercles concentriques, dans le troisième on isole deux points pour retrouver les cercles, dans le quatrième en revanche on est obligé de scinder le plus "grand" des deux



	1	2	3
1	50	0	0
2	0	50	50

1	2	3
47	0	0
1	0	0
1	0	0
1	0	0
0	39	0
0	4	0
0	1	0
0	1	0
0	2	0
0	3	35
0	0	1
0	0	2
0	0	1
0	0	2
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1

Fig. 2.5: Résultats pour les bases de Ruspini et les Iris de Fischer, pour les Iris de Fisher (dimension 4) seuls les tableaux croisés entre "vraies" classes et résultats de la classification sont présentés

cercles en trop de composantes pour sa reconnaissance.

2.4 Limites et améliorations

On remarque ainsi que lorsqu'on définit une distance intra-classes adéquate la classification hiérarchique avec la distance du minimum donne des résultats corrects en classification non supervisée tant du point de vue de la classification elle même que du choix du nombre de classes issu de la lecture du diagramme de distance intra-classes. On peut néanmoins noter quelques limites.

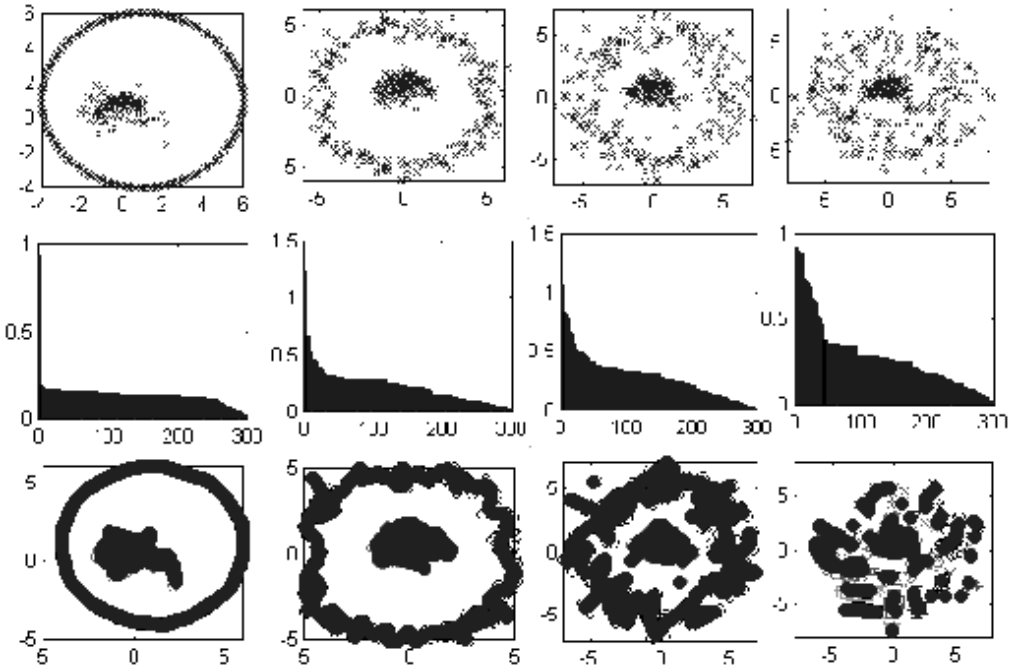


Fig. 2.6: Résultats pour des classes formées par deux cercles concentriques (les classes sont noircies pour plus de lisibilité en noir et blanc)

2.4.1 Cas des classes linéairement séparables

Dans le cas où les classes sont linéairement séparables on obtient de moins bons résultats avec la classification hiérarchique par la distance du minimum qu'avec des méthodes classiques. Ceci s'explique en invoquant le fait que les méthodes classiques, reposant sur l'hypothèse de séparabilité linéaire, hypothèse plus forte que la connexité, telles que les $K-$means, seront meilleures que notre méthode si cette hypothèse sous-jacente est vérifiée. On propose donc de construire un test de séparabilité linéaire des classes, et, si il est vérifié, d'appliquer plutôt les méthodes "classiques". Une manière simple de mettre en place ce test est la suivante :

- Par la classification hiérarchique avec la distance du minimum on obtient un ensemble de classes ;
- Si une analyse discriminante linéaire permet de retrouver les classes de la classification hiérarchique par la distance du minimum, c'est que celles-ci sont linéairement séparables ;

- Si les classes sont linéairement séparables, on préférera une autre méthode telle que les centres mobiles.

2.4.2 Statistiques sur la rupture de distance intra-classes

Dans le cas de l'utilisation de l'hypothèse de connexité pour la modélisation, ce qui nous intéresse est la validation de l'hypothèse de connexité et non la décomposition en classes connexes. Pour cela il faut pouvoir tester le fait que le nuage de points est connexe, c'est-á-dire qu'aucune des segmentations n'est pertinente. C'est pour cela que nous allons essayer de calculer des statistiques sur la rupture de distance intra-classes afin de savoir si les sauts observés sont significatifs ou non. Ce point fera l'objet du chapitre suivant pour le cas particulier où les classes sont issues d'un tirage uniforme.

2.4.3 Séparabilité des classes

Supposons la vraie classification connue. Cette classification ne pourra être obtenue à partir de la méthode décrite précédemment que si (et seulement si) les classes sont séparables c'est-à-dire si :
$\sup\{d_c(x_i, x_j), C(x_i) = C(x_j)\} > \inf\{d(x_i, x_j), C(x_i) \neq C(x_j)\}$ Dans le cas contraire, on aura l'obligation de scinder une classe pour en isoler une autre. Ceci peut provenir de plusieurs phénomènes décrits ci-dessous.

- il peut exister des points liants deux classes
- si les classes sont trop hétérogènes en dispersion, on risque d'être obligé de scinder la classe la plus "dispersée" pour la séparer des autres

Si le premier point n'a pas pu être traité, le second point, plus réaliste, fait l'objet des deux derniers chapitres de la partie classification.

3. VERS UN TEST DE CONNEXITÉ

3.1 Introduction : lien entre la classification et la théorie des graphes

Le lien entre Classification Hiérarchique avec la distance du minimum et le Minimal Spanning Tree (MST) a été souligné de nombreuses fois dans la littérature (initialement par Gower et Ross en 1969 [MST11]). On peut associer à la classification hiérarchique avec la distance du minimum, un graphe en définissant des liaisons (ponts) entre les points de la manière suivante : A chaque étape de la classification prise dans le sens ascendant, on relie les deux points réalisant le minimum de la distance (du minimum) entre les classes.

Au départ tous les points sont séparés (N classes de singletons). On lie les points i_0 et i_1 tels que $d(x_{i_0}, x_{i_1}) = \min(d(x_i, y_j, i \neq j))$ sur le graphe et à cette étape, la classification est $(\{x_{i_0}, x_{i_1}\}, \{x_{i_2}\}, ..., \{x_{i_N}\})$, soit : $(C_1^2, .., C_{N-1}^2)$. On passe de l'étape en $k+1$ classes à celle en k classes en ajoutant une liaison de la manière suivante : si $(i_0, i_1) = argmin(d(C_i^{k+1}, C_j^{k+1}))$.

Si, de plus : $(j_0, j_1) = argmin(d(x_i, x_j), x_i \in C_{i_0}^{k+1}, x_j \in C_{i_1}^{k+1}))$, alors on relie j_0 et j_1 sur le graphe et on agrège $C_{i_0}^{k+1}$ et $C_{i_1}^{k+1}$ dans la classification.

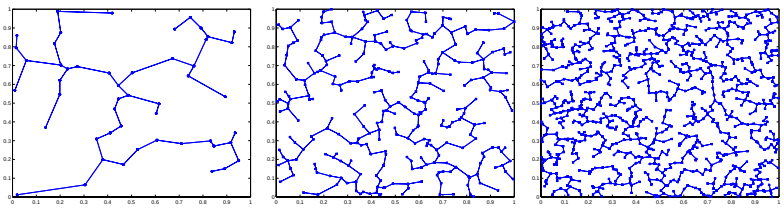


Fig. 3.1: Observations et MST associé pour des tirages uniformes en dimension 2 (50, 300 et 1000 observations)

Un tel graphe est composé de $N-1$ liaisons et tous les points sont liés entre eux par le chemin sur lequel le plus petit des plus grands sauts est atteint, donc sur le chemin permettant de lire la distance connexe d_c entre tous les couples de points. Ces deux considérations font de ce graphe un arbre couvrant de l'ensemble de points qu'on nommera ici MST (pour "Minimum Spanning Tree" au sens où il minimise la distance connexe entre tous les points).

Des statistiques sur les distances des liaisons de ce graphe seront donc source d'information pour étudier la validité de la classification. Indiquons par exemple que :

- La plus grande des distances représente le seuil minimal de connexité des données

- La statistique d'ordre sur la loi des distances indique la loi du nombre de composantes connexes à un seuil donné.

Il a été prouvé dans la littérature ([MST11]]) que le MST tel que nous l'avons défini ici est bien le MST au sens "classique", c'est-à-dire l'arbre couvrant minimal au sens de la somme des longueurs de liaison.

3.2 Les statistiques sur les longueurs des liaisons sur le MST

De nombreux travaux sur la convergence des longueurs de liaisons sur le Minimal Spanning Tree, reposant sur le travail de Beardwood, Halton et Hammersley en 1959 ([MST3]), se concentrent sur le comportement asymptotique de la longueur moyenne d'une liaison : $L_N^d(k) = \frac{1}{N-1}\sum \delta_i^k$ où les δ_i sont les longueurs des liaisons sur le MST calculées sur un tirage uniforme de N points dans $[0,1]^d$. Certains travaux relativement récents ([MST1],[MST2],[MST4],[MST5] entre 1992 et 1996) montrent l'existence d'un théorème central limite pour $L_N^d(k)/N^{k/d}$. Plus récemment encore Penrose et Yudich, en 2003 ([MST9]) ont, dans une certaine mesure, élargi le résultat à des tirages non uniformes. De telles statistiques, portant sur les moments empiriques des longueurs sur 1 tirage, ne nous sont pas utiles pour établir un test de connexité. En effet si on s'intéresse aux différentes valeurs prises par δ_i, en considérant uniquement leurs moment empirique, l'information est trop résumée. Ces travaux nous ont néanmoins donné une indication sur la vitesse de convergence vers 0 (en $N^{1/d}$) des longueurs des liaisons, et donc sur la

bonne normalisation.

En plus du théorème central limite, Penrose a montré dans [MST6] (1997), que si M_N est la plus grande longueur sur le MST d'un tirage uniforme, la loi de $N\pi_d M_N^d$ converge faiblement vers une double exponentielle (avec π_d le volume d'une boule unité en dimension d). Ce résultat est valable lorsque le tirage est uniforme sur un cube en dimension inférieure ou égale à 2 ou sur un tore en dimension supérieure. Un résultat similaire a été montré dans [MST7] (1998) pour des tirages gaussiens.

Un tel résultat est un premier pas vers la construction d'un test. Malheureusement une indication sur le seul maximum n'est pas suffisante par exemple lorsque la coupure s'effectue au delà de deux classes. On pourrait aisément résoudre ce problème en itérant de manière hiérarchique, mais la principale limitation vient de la topologie torique pour des dimensions supérieures à 2. Cependant on trouve dans [MST6] de nombreuses idées intéressantes.

Enfin dans [MST8] (1996), Penrose a montré la convergence de certaines statistiques sur le MST non seulement lorsque le nombre de points tend vers l'infini mais aussi lorsque la dimension tend vers l'infini. Ici nous n'établirons pas de tels résultats mais c'est une voie intéressante. En effet on n'a pas pu identifier par le calcul les lois limites (ni les reconnaître sur les graphiques). Ainsi avoir une convergence avec la dimension nous éviterait d'avoir à construire un nombre de tables infini.

Si il peut sembler peu réaliste, en pratique, de prendre comme hypothèse des tirages uniformes l'intérêt des comparaisons, pour des classifications, entre tirages uniformes et observation a été souligné, et exploitée dans [HIE12].

3.3 Résultats empiriques

On a observé la fonction de répartition empirique des $n-1$ valeurs de $n^{1/d}\delta_n(X)$ pour des tirages de 10, 50, 500 et 1000 points suivant des lois uniformes sur $[0,1]^d$ (5 tirages par nombre de points). On observe bien la convergence des fonctions de répartitions empiriques, qui semble être presque sûre, ce qu'on ne démontrera pas dans la partie suivante (on ne montrera que la convergence en moyenne de la fonction de répartition empirique).

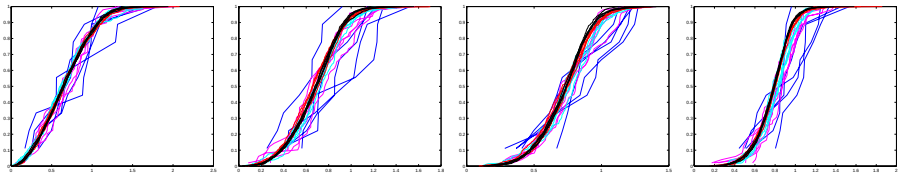


Fig. 3.2: Fonction de répartition empirique des longueurs de liaison pour des dimensions 2 à 5 pour (10, 50 200, 500 et 1000 points) respectivement (bleu, magenta, cyan, rouge et noire)

A titre indicatif on donne aussi des fonctions de répartitions empiriques pour des tirages de 1000 points en fonction de la dimension pour des dimensions 2 à 10 et la dimension 20.

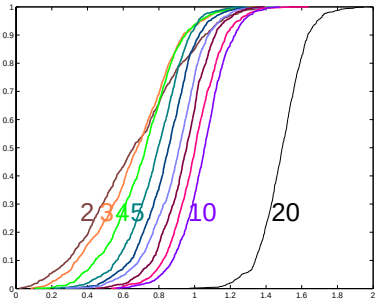


Fig. 3.3: Fonctions de répartition empiriques des longueurs de liaison en fonction de la dimension

On n'observe pas de convergence en ne pondérant que par $n^{1/d}$ mais les courbes semblent effectivement de plus en plus semblables à une translation près. En ramenant la moyenne des longueurs de liaison en 0, il semble y avoir convergence des fonctions de répartition empiriques avec la dimension.

3.4 Existence d'un comportement asymptotique

Dans un premier temps on va mettre en place des éléments visant à montrer que la loi d'une distance tirée au hasard sur un MST et pondérée en fonction du nombre de points et de la dimension (par $N^{1/d}$) converge vers une certaine loi (qu'on ne sait pas aujourd'hui expliciter autrement que par simulation). Pour cela on va tout d'abord étudier le comportement asymptotique des différents moments de $n^{1/d}\delta_n$ où δ_n

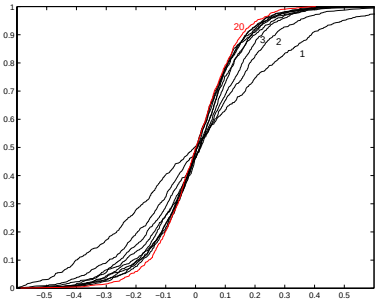


Fig. 3.4: Fonctions de répartition empiriques, ramenée en moyenne à 0 en fonction de la dimension

représente la variable aléatoire suivante :

- On tire un $n-$échantillon uniformément sur $[0,1]^d$.
- On construit le $MST(X)$ et on tire au hasard une de ses liaisons, dont on note δ_n la longueur de cette liaison.

Calculs préliminaires

Lemme 3.4.1: Si X' est un sous échantillon de X ($X' \subset X$).

Si x_i et x_j sont dans X'

Si $[x_i, X_j]$ est une liaison du $MST(X)$

Alors $[x_i, X_j]$ est une liaison du $MST(X')$

Autrement dit : le graphe de la restriction contient la restriction du graphe.

La démonstration est évidente en utilisant la caractérisation de la distance entre les points : si deux points x_1 et x_2 sont liés sur G, c'est que $d(x_1, x_2) = min(sautmax(che(X))) \leq min(sautmax(che(X')))$, l'autre inégalité étant immédiate.□

Lemme 3.4.2: La longueur d'une liaison sur le MST d'un tirage uniforme sur $[0,1]^d$ est bornée par $\sqrt{(}d)$.

Là démonstration est aussi évidente, étant donné que la longueur maximum d'une liaison sur un hyper-cube $[0,1]^d$ est bornée par $\sqrt{(}d).\square$

Lemme 3.4.3: Soit $\delta(N,X)$ la longueur d'une liaison tirée au hasard sur le $MST(X)$ où X est un $N-$échantillon uniforme sur $[0,1]^d$

- $\forall u \in]0;1/d[$ $P(\delta(N,X) > N^{-1/d+u})$ tend vers 0 lorsque N tend vers ∞
- $\mathbb{E}_X(\delta^k(N,X)N^{1/d-u})$ tend vers 0 lorsque N tend vers ∞

Supposons que A et B distants de x soient tels que $[A,B]$ soit une liaison du $MST(X)$. Alors nécessairement, l'intersection de la boule de rayon x et de centre A et de la boule de centre B et de rayon x est vide. Sinon il existerait au moins un point M tel que $d(A,M) < d(A,B)$ et $d(B,M) < d(A,B)$, ce qui est impossible. Le volume de l'intersection des deux boules est proportionnel à x^d, on notera ν_d la constante de proportionnalité.

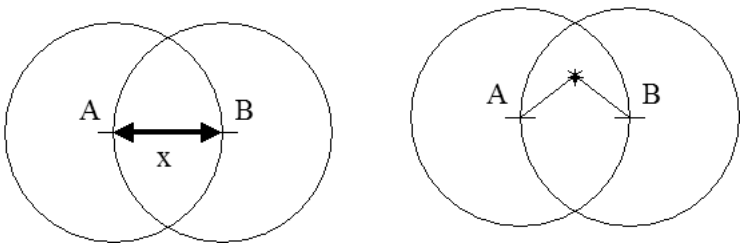


Fig. 3.5:

Ainsi :
$P(\delta(N,X)) \geq x) \leq N(1-\nu_d x^d)^{N-1}$
et :
$P(\delta(N,X)) \geq N^{-1/d+u}) \leq N(1-\nu_d N^{-1+du})^{N-1}$
avec $N(1-\nu_d N^{1-du})^{N-1} = Nexp((N-1)ln(1-\nu_d N^{-1+du}))$
si $u \in]0,1/d[$ alors $N(1-\nu_d N^{1-du})^{N-1} \sim Nexp(-\nu_d N^{du}) \to 0.\square$

Pour passer à la propriété sur les moments :
Si $u \in]0,1/d[$ alors $\exists v > 0$ tel $]u-v[\in]0,1/d[$ et, pour un tel v :

$$P(\delta(N,X)N^{1/d-u} > N^{-v}) \leq N(1-\nu_d N^{-1+(u-v)d})^{N-1} \sim Ne^{-\nu_d N^{(u-v)/d}}$$

comme :

$$\mathbb{E}((\delta(N,X)N^{1/d-u})^k) = \int_0^{N^{-v}} x^k f(x)dx + \int_{N^{-v}}^{\sqrt{d}N^{1/d-u}} x^k f(x)dx$$

$$\mathbb{E}((\delta(N,X)N^{1/d-u})^k) < N^{-vk} + (\sqrt{d}N^{1/d-u})^k N(1-\nu_d N^{-1+(u-v)d})^{N-1}$$

Le second terme est négligeable devant le premier et :
$\forall v < u\ \mathbb{E}((\delta(N,X)N^{1/d-u})^k) = \theta(N^{-vk})$
D'où $\mathbb{E}((\delta(N,X)N^{1/d-u})^k) \to 0$. □

Lemme 3.4.4: Le MST n'est pas inclus dans un graphe des $n-$plus proches voisins avec une probabilité tendant vers 0 géométriquement en n

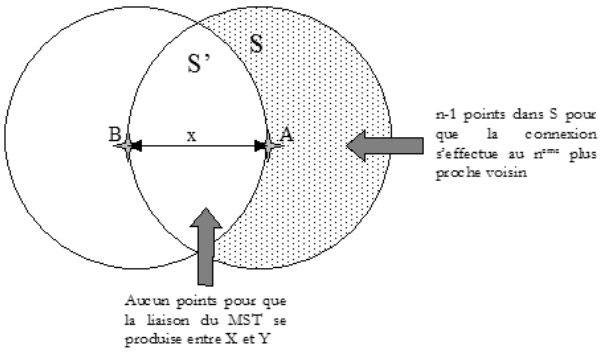


Fig. 3.6: Evaluation de la probabilité que l'on aille jusqu'au $n^{ième}$ voisin sur le MST

Pour qu'un point A soit connecté sur le MST à son $n^{ième}$ voisin et que cette connexion ait lieu avec une longueur x (et relie A à B) il faut, d'une part qu'il y ait au moins $k-1$ points dans la boule $\mathcal{B}(A,x)$ (pour garantir que la liaison de longueur x est la $n^{ième}$ et d'autre part il ne faut pas qu'il y ait de points à une distance inférieure à x de A et B.

Soit $\lambda_d = \frac{V(\mathcal{B}(A,x))}{V(\mathcal{B}(A,x)\setminus\mathcal{B}(B,x))}$

Sachant x : la probabilité d'avoir $k-1$ points dans S et aucun point dans S' une fois A placé est :

$$C_{N-1}^{k-1}(\lambda_d \pi_d x^d)^{k-1}(1-\pi_d x^d)^{N-k}$$

Le maximum de cette fonction a lieu pour $x^d = \frac{k-1}{(N-1)\pi_d}$ et la probabilité d'avoir une liaison du MST correspondant à un $k-$ plus proche voisin est inférieure à:

$$(\lambda_d)^{n-1} C_{N-1}^{n-1} \frac{(n-1)^{n-1}(N-n)^{N-n}}{(N-1)^{N-1}}$$

D'après la formule de Stirling les quantités $\frac{n!}{(n/e)^n\sqrt{n}}$ et son inverse sont bornées. Comme, de plus $\frac{\sqrt{N-1}}{\sqrt{(k-1)(N-k)}}$ est aussi une quantité bornée, il existe alors une constante c_1 telle que la probabilité de ne pas être inclus dans le graphe des $n-$ plus proches voisins soit inférieure à : $c_1(\lambda_d)^{k-1}$.□

Lemme 3.4.5: Soient :

- X un $N-$échantillon uniforme sur $[0,1]^d$
- Φ la fonction de répartition de la loi normale centrée réduite
- $\mu \in]0,1[$
- $a \in]0,\mu[$
- $X' = X|_{[0,\mu]^d}$
- G_k le graphe des $k-plus$ proches voisins de X
- et G'_k le graphe des $k-plus$ proches voisins de X'

La probabilité qu'une liaison de G'_k partant d'un point de $[0,1-a]^d$ ne soit pas une liaison de G_k tend vers 0 quand N tend vers l'infini (avec une vitesse d'au plus : $\Phi\left(\frac{k-2(a/2)^d\pi_d N}{\sqrt{N2(a/2)^d\pi_d(1-2(a/2)^d\pi_d)}}\right)$)

Pour qu'une liaison de G'_k partant d'un point de $[0,1-a]^d$ ne soit pas une liaison de G_k il faut et il suffit qu'un de ses $k-$plus proches voisins (dans X) soit dans $[0,1]^d \backslash [0,\mu]^d$. Pour cela, la situation la plus probable (illustrée sur la figure 3-7) est de probabilité :

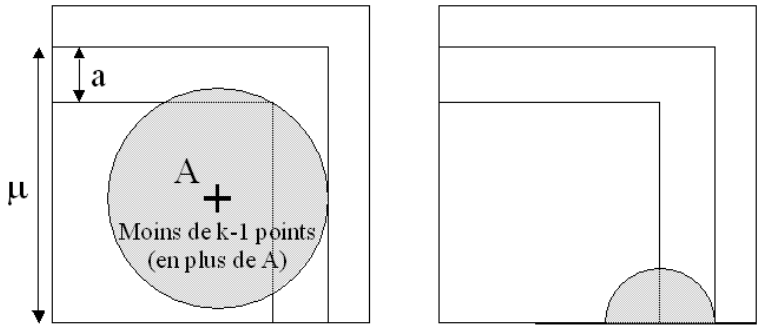


Fig. 3.7: Evaluation de la probabilité qu'une liaison du graphe de la restriction ne soit pas une liaison de la restriction du graphe

$$\sum_{n=0}^{k} C_N^k (\pi_d a^d / 2^{d-1})^k (1 - \pi_d a^d / 2^{d-1})^{N-k}$$

et, en utilisant l'approximation gaussienne de la loi binômiale la dernière expression est équivalente à :

$$\Phi\left(\frac{k - 2(a/2)^d \pi_d N}{\sqrt{N 2(a/2)^d \pi_d (1 - 2(a/2)^d \pi_d)}}\right) \square$$

Corollaire 3.4.1: Soient :

- X un $N-$échantillon uniforme sur $[0,1]^d$
- $\mu \in]0,1[$
- $a \in]0,\mu[$
- $X' = X|_{[0,\mu]^d}$
- G le MST de X
- G' le MST de X'

La probabilité qu'une liaison de G' tirée au hasard dans $[0, \mu - a]^d$ ne soit pas une liaison de G tend vers 0 quand N tend vers l'infini.

Soit un nombre de voisins k qu'on va tout d'abord supposé fixé. Si les MST G et G' sont inclus dans les graphes des $k-$plus proches voisins

de X et X', alors une liaison de G' tirée dans $[0, \mu - a]^d$ est une liaison de G avec une probabilité supérieure $1 - \Phi\left(\frac{k-2(a/2)^d\pi_d N}{\sqrt{N2(a/2)^d\pi_d(1-2(a/2)^d\pi_d)}}\right)$.

Majorons la probabilité que ni G ni G' ne soient inclus dans les graphe des $k-$ plus proches voisins :

$P((G' \subsetneq G'_k) \cap (G \subsetneq G_k)) < max(P(G \subsetneq G_k), P(G' \subsetneq G'_k)) = c_1\lambda_d^{k-1}$

Par suite la probabilité qu'une liaison de G' ne soit pas une liaison de G est inférieure à :

$$p_1(d,k,a,N) = c_1\lambda_d^{k-1} + (1 - c_1\lambda_d^{k-1})\Phi\left(\frac{k - 2(a/2)^d\pi_d N}{\sqrt{N2(a/2)^d\pi_d(1-2(a/2)^d\pi_d)}}\right).$$

Cette inégalité est vraie quelque soit k.

En choisissant par exemple : $k = N^\beta$ avec $\beta \in]0,1[$ on a :

$$p_1^*(d,\beta,a,N) = c_1\lambda_d^{N^\beta - 1} + \Phi\left(\frac{N^\beta - 2(a/2)^d\pi_d N}{\sqrt{N2(a/2)^d\pi_d(1-2(a/2)^d\pi_d)}}\right) \to 0.\square$$

Dans la suite on prendra une marge a décroissante en N de manière à ce que $(a(N)/2)^d\pi_d N = N^\beta$ soit, $a(N) = 2(1/\pi_d)^{1/d}N^{\frac{\beta-1}{d}}$.

Pour cette marge on aura :

$$p_1^o(d,\beta,N) = O((c_1/\lambda_d)\lambda_d^{N\beta} + \Phi\left(-\frac{N^{\beta/2}}{\sqrt{2}}\right)$$

et, au total :

$$p_1^o(d,\beta,N) = O(\rho^{N^\beta})$$

Corollaire 3.4.2: **Equation de récurrence sur les moments** Soit $u_N(k)$ le moment d'ordre k de δ_N. Alors $u_N(k)$ vérifie l'équation de récurrence suivante :

$$|u_N(k) - p^{d-k}\sum_{n=2}^{N} C_N^n \frac{n-1}{N-1}\left(\frac{1}{p^d}\right)^n\left(1-\frac{1}{p^d}\right)^{N-n} u_n(k)| \leq$$

$$O(p^d/N)u_N(k) + O(N^{\frac{\beta-1}{d}})u_N^*(k) + O(C_d\rho_d^{N^\beta})$$

avec $\forall u > 0\ N^{(k/d)-u}u_N^*(k) \to 0$

On divise l'hypercube $[0,1]^d$ en p^d hypercubes C_i de coté $1/p$ et on note $G_p(X)$ l'union des $MST(X|_{Ci})$. On note D_i l'hypercube C_i auquel on a ôté les marges d'épaisseur $a(N)$

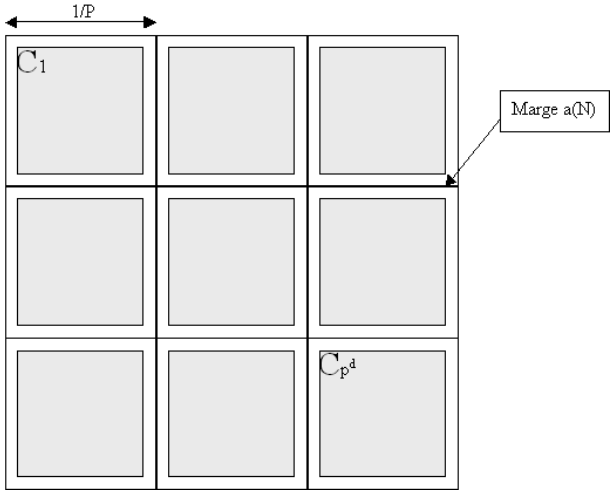


Fig. 3.8: mise en équation des moments

Une liaison du $MST(X)$ n'a pas d'extrémité dans un des D_i avec une probabilité : $q_1 \sim 1 - p^d(1/p - 2a(N))^d = 1 - (1 - 2a(N)p^d)^d \sim 2da(N)p^d = O(N^{\frac{\beta-1}{d}})$.
Si une liaison du $MST(X)$ a une extrémité dans l'un des D_i alors elle n'appartient pas à $G_p(X)$ avec une probabilité $q_2 = O(C_d\rho_d^{N\beta})$.

On suppose dans la suite $N >> p^{\frac{d}{1-\beta}}$ ce qui correspond au fait que la division en hyper-cubes est possible en laissant des domaines, à l'intérieur des marges, suffisamment grands.

On note :

- u_N l'espérance, sur X, de la longueur d'une liaison tirée au hasard sur le $MST(X)$
- u_N^* l'espérance, sur X de la longueur d'une liaison tirée au hasard sur les $MST(X)$ dont aucune extrémité n'est dans un des D_i. Comme les seules conditions sont des conditions de positions des extrémités, les calculs du lemme 4.4.3 s'appliquent et $\forall u > 0$ $N^{1/d-u}u_N^* \to 0$

- u_N^{**} l'espérance, sur X de la longueur d'une liaison du $MST(X)$ dont une extrémité est dans l'un des D_i mais qui n'appartient pas à $G_p(X)$. Ici les conditions sont plus fortes et on peut seulement affirmer que $u_N^{**} < \sqrt(d)$
- w_N l'espérance, sur X, de la longueur d'une liaison tirée au hasard sur le $G_p(X)$

On a alors :

$$u_N = q_1 u_N^* + (1-q_1)q_2 u_N^{**} + (1-q_1)(1-q_2)w_N$$

et ainsi :

$$|u_N - w_N| < q_1 u_N^* + (1-q_1)q_2 u_N^{**}$$

et :

$$|u_N - w_N| < q_1 u_N^* + q_2 u_N^{**}$$

On va désormais calculer w_N : on a :

$$w_N = \sum_{\sum_{i=1}^{p^d} n_i = N} \sum_{i=1}^{p^d} \frac{max(n_i - 1, 0)}{N^*(n_1, .., n_{p^d})} \frac{u_{n_i}}{p} \frac{n!}{\pi(n_i!)} \left(\frac{1}{p^d}\right)^n$$

avec : $N^*(n_1, .., n_{p^d}) = \sum(max(n_i - 1, 0))$ Pour obtenir cette formule on a conditionné par la répartition du nombre de points dans chacun des C_i (qui suit une loi multinomiale). Si on a n_i points dans C_i on a $n_i - 1$ liaisons dans le MST calculé sur ces points (si $n_i > 0$) et n_i/N^* représente ainsi la probabilité de tirer la liaison parmi celles du $MST(C_i)$. Enfin l'espérance des longueurs des liaisons sur le $MST(C_i)$ est en loi, au facteur d'échelle $1/p$ près, identique à celles des liaisons tirées sur $[0,1]^d$.

On a, de manière évidente : $N - p^d \leq N^*(n_1, .., n_{p^d}) < N$ car $n_i - 1 \geq max(n_i - 1, 0) < n_i$
donc : $N - p^d \leq N^*(n_1, .., n_{p^d}) <\leq N - 1$ et ainsi :

$$w_N = (1 + O(p^d/N)) \sum_{\sum_{i=1}^{p^d} n_i = N} \sum_{i=1}^{p^d} \frac{max(n_i - 1, 0)}{N-1}) \frac{u_{n_i}}{p} \frac{n!}{\pi(n_i!)} \left(\frac{1}{p^d}\right)^n$$

On a de plus :

$$\sum_{\sum_{i=1}^{p^d} n_i = N} \sum_{i=1}^{p^d} \frac{max(n_i - 1, 0)}{N-1}) \frac{u_{n_i}}{p} \frac{n!}{\pi(n_i!)} \left(\frac{1}{p^d}\right)^n =$$

$$p^{d-1}\sum_{n=0}^{N} C_N^n \frac{max(n-1,0)}{N-1}\left(\frac{1}{p^d}\right)^n\left(1-\frac{1}{p^d}\right)^{N-n} u_n =$$

$$p^{d-1}\sum_{n=2}^{N} C_N^n \frac{n-1}{N-1}\left(\frac{1}{p^d}\right)^n\left(1-\frac{1}{p^d}\right)^{N-n} u_n$$

On obtient ainsi, au total :

$$|u_N - p^{d-1}\sum_{n=2}^{N} C_N^n \frac{n-1}{N-1}\left(\frac{1}{p^d}\right)^n\left(1-\frac{1}{p^d}\right)^{N-n} u_n| \leq$$

$$O(p^d/N)u_N + O(N^{\frac{\beta-1}{d}})u_N^* + O(C_d\rho_d^{N^\beta})$$

avec $\forall u > 0\ N^{1/d-u}u_N^* \to 0$

De manière parfaitement analogue, si $u_N(k)$ représente le moment d'ordre k de la taille d'une liaison tirée aléatoirement sur un MST sur un $n-$échantillon uniforme sur $[0,1]^d$ on obtiendra :

$$|u_N(k) - p^{d-k}\sum_{n=2}^{N} C_N^n \frac{n-1}{N-1}\left(\frac{1}{p^d}\right)^n\left(1-\frac{1}{p^d}\right)^{N-n} u_n(k)| \leq$$

$$O(p^d/N)u_N(k) + O(N^{\frac{\beta-1}{d}})u_N^*(k) + O(C_d\rho_d^{N^\beta})$$

avec $\forall u > 0\ N^{(k/d)-u}u_N^*(k) \to 0$

Le moment sur G_p reste la somme des moments par indépendance des tirages sur chaque sous-cube C_i. En revanche le facteur d'échelle pour passer d'un cube de coté $1/p$ à un cube de coté 1, est pour le moment d'ordre k : $(1/p)^k$. □

Lemme 3.4.6: Soient μ et λ deux réels positifs avec de plus μ inférieur à 1.

Soit $\alpha = ln(\lambda)/ln(\mu)$

Soit $R(n,\alpha,\mu)$ défini de la manière suivante :

$$R(n,\alpha,\mu) = \frac{\Gamma(n+1+\alpha)}{\Gamma(n+1)} \int_{\mu}^{1-\mu} (1-\mu-t)^n t^{\alpha-1}$$

Soit $v_n(\mu,\lambda)$ définie par récurrence de la manière suivante :
$v_0(\mu,\lambda)$ est un réel quelconque et :
$\forall n : v_n(1-R(n,\alpha,\mu)) = \lambda \sum_{k=0}^{n} C_n^k \mu^k (1-\mu)^{n-k} v_k$
Alors $v_n = v_0 \frac{\Gamma(n+1+\alpha)}{\Gamma(n+1)}$

En effet, en temps que solution d'une équation de récurrence linéaire, une telle suite est définie de manière unique à partir de son premier terme dès que pour tout n : $R(n,\alpha,\mu) + \mu^n \neq 1$ (ce qu'on supposera vérifié) car l'équation de récurrence s'écrit aussi :

$$v_n(1-R(n,\alpha,\mu)-\lambda\mu^n) = \lambda \sum_{k=0}^{n-1} C_n^k \mu^k (1-\mu)^{n-k} v_k.$$

D'après le développement de Taylor avec reste intégral de $f(x) = x^{n+\alpha}$ on à :

$$1 = (\mu+(1-\mu))^{n+\alpha} = \sum_{k=0}^{n} \frac{\Gamma(n+\alpha+1)}{\Gamma(n+\alpha+1-k)(k!)} \mu^{n+\alpha-k}(1-\mu)^k + R(n,\alpha,\mu)$$

D'où :

$$(1-R(n,\alpha,\mu))\frac{\Gamma(n+1)}{\Gamma(n+1+\alpha)} v_0 = \mu^{\alpha} \sum_{k=0}^{n} C_n^k \mu^k (1-\mu)^{n-k} \frac{\Gamma(k+1)}{\Gamma(k+1+\alpha)} v_0$$

Ainsi, si $\mu^{\alpha} = \lambda$, soit pour $\alpha = ln(\lambda)/ln(\mu)$ on obtient très exactement :
$v_n = v_0 \frac{\Gamma(n+1)}{\Gamma(n+1+\alpha)} \sim v_0 n^{-\alpha}$ □

Lemme 3.4.7 (Retour a l'équation sur les moments): Les moments $u_n(k)$ sont solutions d'une équation de récurrence linéaire dont les solutions de l'équation homogène associée sont équivalentes (lorsque n tend vers l'infini) à $n^{-k/d}$

On avait :

$$|u_N(k) - p^{d-k} \sum_{n=2}^{N} C_N^n \frac{n-1}{N-1} \left(\frac{1}{p^d}\right)^n \left(1 - \frac{1}{p^d}\right)^{N-n} u_n(k)| \leq$$
$$O(p^d/N)u_N(k) + O(N^{\frac{\beta-1}{d}})u_N^*(k) + O(C_d \rho_d^{N^\beta})$$

avec $\forall u > 0\ N^{(k/d)-u} u_N^*(k) \to 0$
Si on note $v_n = (n-1)u_n$, on obtient :

$$|v_N(k) - p^{d-k} \sum_{n=2}^{N} C_N^n \left(\frac{1}{p^d}\right)^n \left(1 - \frac{1}{p^d}\right)^{N-n} v_n(k)| \leq$$
$$O(p^d/N)v_N(k) + O(N^{\frac{\beta-1}{d}})Nu_N^*(k) + NO(C_d \rho_d^{N^\beta})$$

Et ainsi :

$$|v_N(k)(1-R(N,(-1+k/d),(1/p)^d)) - p^{d-k} \sum_{n=0}^{N} C_N^n \left(\frac{1}{p^d}\right)^n \left(1 - \frac{1}{p^d}\right)^{N-n} v_n(k)| \leq$$
$$v_N(k)R(N,(-1+k/d),(1/p)^d) + p^{d-k}\left(1 - \frac{1}{p^d}\right)^N v_0 + Np^{d-k}\left(1 - \frac{1}{p^d}\right)^{N-1} v_0$$
$$+O(p^d/N)v_N(k) + O(N^{\frac{\beta-1}{d}})Nu_N^*(k) + ON(C_d \rho_d^{N^\beta})$$

On peut donc écrire que :

$$v_N(k)(1 - R(N,(-1+k/d),(1/p)^d)) =$$
$$p^{d-k} \sum_{n=0}^{N} C_N^n \left(\frac{1}{p^d}\right)^n \left(1 - \frac{1}{p^d}\right)^{N-n} v_n(k) + \varepsilon(N)$$

avec $\varepsilon(N) = o(N^{1-(k/d)-u})$ pour tout $u \in [0, (\beta-1)/2d[$ (dans tous les termes de la majoration de l'erreur, c'est l'avant dernier qui tend vers 0 le plus lentement).

On a montré dans le lemme précédent que les solutions de l'équation homogène sont en $c\frac{\Gamma(N)}{\Gamma(N-1+k/d)} \sim cN^{1-k/d}$.

Enfin comme $v_n(k) = (n-1)u_n(k)$, on obtient, pour les solutions de l'équation homogène sur les moments, des suites proportionnelles à : $c\frac{\Gamma(N)}{\Gamma(N-1+k/d)(N-1)} \sim cN^{-k/d}$.□

Théorème 3.4.1: Les moments d'ordre k $u_n(k)$ vérifient : $u_n(k)n^{k/d}$ admet une limite finie.

On va continuer à travailler sur $v_n(k) = (n-1)u_n(k)$. On a déjà montré que les v_n étaient solutions d'une équation récurrente linéaire avec second membre dont les solutions de l'équation homogène ont le comportement voulu. On va désormais montrer qu'il existe une solution particulière négligeable devant toute solution générale (non nulle).

Pour passer aux solutions de l'équation générale, on simplifie dans un premier temps l'écriture :

$$v_n(k) = \sum_{i=1}^{n-1} a(n,k)v_i(k) + \varepsilon(n)$$

avec dans notre cas :

$$a(n,k) = \frac{p^{d-k}C_n^k(1/p^d)^k(1-(1/p^d))^{n-k}}{1-R(n,1-k/d,(1/p)^d)-p^{d-k}p^n}$$

Ceci peut aussi s'écrire :

$$v_n(k) = b_n v_0(k) + \sum_{i=1}^{n} \alpha(n,i)\varepsilon(i)$$

avec b_n et $\alpha(n,k)$ qui vérifient les équations de récurrences suivantes :

- $b_{n+1} = \sum_{j=0}^{n} a(n,j)b_j$
- $\alpha(n+1,n+1) = 1$
- $\forall j < n+1\ \alpha(n+1,j) = \sum_{i=j}^{n} a(n+1,i)\alpha(i,j)$

D'après ce qu'on a montré précédemment, le terme en $b_n v_0$ correspond aux solutions de l'équation sans second membre $b_n = \frac{\Gamma(n)}{\Gamma(n-1+k/d)}$.

Il reste désormais a montrer, pour conclure, que :

$$\sum_{j=1}^{n} \alpha(n,j)\varepsilon(j)$$

a un comportement négligeable devant celui des solutions de l'équation sans second membre (i.e. devant $\frac{\Gamma(n)}{\Gamma(n-1+k/d)}$).

Pour cela on travaille, dans un premier temps sur les coefficients α. On montre par récurrence que les coefficients α ont un comportement similaire aux coefficients a

On a :

- $\alpha(n+1,n+1) = 1$
- $\forall j < n+1 \ \alpha(n+1,j) = \sum_{i=j}^{n} a(n+1,i)\alpha(i,j)$

Pour simplifier les notations on écrit :

- $a(n,k) = \frac{\lambda C_n^k(\mu)^k(1-\mu)^{n-k}}{1-r(n)} avec : \lambda = p^{d-k}$, $\mu = (1/p^d)$, $r(n) = R(n,(1-k/d),1/p^d)$

Ainsi, on montre, par récurrence que :

- Si on écrit : $\alpha(n+k,n) = a(n,k)(1+Q_n(k))$
- Alors $Q_n(k) = \lambda\mu^n \sum_{i=1}^{k-1}([C_k^i\mu^i]/[1-r(n+i)])(1+Q_n(i))$
- Et, par récurrence $Q_n(k) = \lambda\mu^n(1+\mu)^k(1+o_n(\lambda\mu^n(1+\mu)^k)$

Alors :

$$\sum_{j=1}^{n} \alpha(n,j)\varepsilon(j) =$$

$$\varepsilon(n)+\sum_{j=1}^{n-1} a(n,j)\varepsilon(j)+\sum_{j=1}^{n-1} a(n,j)\lambda\mu^j(1+\mu)^{n-j}(1+o_j(\lambda\mu^j(1+\mu)^{n-j})\varepsilon(j)$$

D'où on tire l'existence d'une constante M telle que :

$$\sum_{j=1}^{n} \alpha(n,j)\varepsilon(j) =\leq \varepsilon(n)+\sum_{j=1}^{n-1} a(n,j)\varepsilon(j)+M\sum_{j=1}^{n-1} a(n,j)\lambda\mu^j(1+\mu)^{n-j}\varepsilon(j)$$

En utilisant le fait que $\varepsilon(N) = o(N^{1-(k/d)-u})$ pour tout $u \in [0, (\beta - 1)/2d[$.

On peut écrire que $\varepsilon(i) \leq m_u \frac{\Gamma(i+1)}{\Gamma(i+(k/d)+u)}$ pour tout $u \in [0, (\beta-1)/2d[$.

Et :

$$\sum_{j=1}^{n} \alpha(n,j)\varepsilon(j) \leq$$

$$m_u \frac{\Gamma(n+1)}{\Gamma(n+(k/d)+u)}$$

$$+m_u \sum_{j=1}^{n-1} a(n,j) \frac{\Gamma(j+1)}{\Gamma(j+(k/d)+u)}$$

$$+Mm_u \sum_{j=1}^{n-1} a(n,j)\lambda\mu^j(1+\mu)^{n-j} \frac{\Gamma(j+1)}{\Gamma(j+(k/d)+u)}$$

Le premier terme de l'inéquation est négligeable devant :

$$\frac{\Gamma(n+1)}{\Gamma(n+(k/d))}$$

Le deuxième terme de l'inéquation vaut :

$$m_u p^{d-k}(1/p^d)^{1-k/d-u} \frac{\Gamma(i+1)}{\Gamma(i+(k/d)+u)} \frac{1-R(n,(1-(k/d)-u),(1/p^d))}{1-R(n,(1-(k/d)),(1/p^d))}$$

et est ainsi aussi négligeable devant :

$$\frac{\Gamma(n+1)}{\Gamma(n+(k/d))}$$

Enfin, le troisième terme de l'inéquation vaut :

$$\frac{1}{1-R(n,(1-(k/d)),(1/p^d))} Mm_u \sum_{j=1}^{n-1} C_n^j \lambda\mu^{2j}(1-\mu)^{n-j}(1+\mu)^{n-j} \frac{\Gamma(j+1)}{\Gamma(j+(k/d)+u)}$$

$$= \frac{1}{1-R(n,(1-(k/d)),(1/p^d))} Mm_u \sum_{j=1}^{n-1} C_n^j \lambda(\mu^2)^j(1-\mu^2)^{n-j} \frac{\Gamma(j+1)}{\Gamma(j+(k/d)+u)}$$

Soit :

$$Mm_u \frac{1 - R(n, (1 - (k/d) - u), (1/p^{2d}))}{1 - R(n, (1 - (k/d)), (1/p^d))} p^{d-k} (1/p^d)^{1-k/d-u} \frac{\Gamma(n+1)}{\Gamma(n + (k/d) + u)}$$

Et est ainsi aussi négligeable devant :

$$\frac{\Gamma(n+1)}{\Gamma(n + (k/d))}$$

Ainsi, au total on peut écrire que :

$$v_n(k) = c(k) \frac{\Gamma(n+1)}{\Gamma(n+1+k/d)} + o\left(\frac{\Gamma(n+1)}{\Gamma(n+1+k/d)}\right).$$

D'où :

$$u_n(k) = c(k) \frac{\Gamma(n+1)}{\Gamma(n+1+k/d)(n-1)} + o\left(\frac{\Gamma(n+1)}{\Gamma(n+1+k/d)(n-1)}\right)$$

Si $c(k) \neq 0$ $u_n(k) \sim c(k) n^{-k/d}$ soit encore :
$u_n(k) n^{k/d}$ admet une limite finie $c(k, d)$.

Remarque sur la vitesse de convergence : Les calculs précédents montrent que :
$\forall\ u \in [0, (\beta - 1)/2d[$
$C_{u,d}$ une constante ne dépendant que de u et d tel que : $|u_n(k) n^{k/d} - c(k, d)| < C_{u,d} n^{-u}$. □

Ainsi, les moments de $X_n = n^{k/d} \delta_n$ convergent vers une limite finie. Pour avoir une convergence en loi de X_n il faut, maintenant montrer que les moments limites satisfont les conditions de Carleman.

Théorème 3.4.2: $X_n = n^{k/d} \delta_n$ converge en loi

Dans le lemme 3.4.3, on avait montré que :

$$P(\delta_N^k > x) \leq N(1 - \nu_d x^{d/k})^{N-1} dx.$$

Donc

$$\mathbb{E}(\delta_N^k) \leq N \int (1 - \nu_d x^{d/k})^{N-1} dx,$$

dont on tire aisément :

$$\mathbb{E}\left(\delta_N^k \frac{\Gamma(N+k/d)}{\Gamma(N)}\right) \leq N\left(\frac{1}{\nu_d}\right)^{k/d} \frac{k}{d}\Gamma(k/d)$$

Ainsi, en utilisant la remarque sur la vitesse de convergence des moments, les équivalents des fonctions Γ et cette dernière inégalité, on obtient :

$\forall n$; $c(k,d) \leq C_{u,d} n^{-u} + n\left(\frac{1}{\nu_d}\right)^{k/d} \frac{k}{d}\Gamma(k/d)$

En recherchant le minimum en n on obtient ainsi :

$$c(k,d) \leq C_{u,d}^{u/(u+1)} \left(u^{-u/(u+1)+u^{1/u+1}}\right) \left(\left(\frac{1}{\nu_d}\right)^{k/d} \frac{k}{d}\Gamma(k/d)\right)^{\left(\frac{u}{u+1}\right)}.$$

Comme, $\left(\frac{1}{\nu_d}\right)^{k/d} \frac{k}{d}\Gamma(k/d)$ tend vers l'infini, on travaillera, sans plus le préciser par la suite, pour les n "suffisamment" grands pour que $\left(\left(\frac{1}{\nu_d}\right)^{k/d} \frac{k}{d}\Gamma(k/d)\right)$ soit supérieur à 1 :

Alors :

$$c(k,d)^{1/k} \leq [C_{u,d}^{u/(u+1)}]^{1/k} \left(u^{-u/(u+1)+u^{1/u+1}}\right)^{1/k} \left(\frac{1}{\nu_d}\right)^{1/d} \left(\frac{k}{d}\Gamma(k/d)\right)^{1/k}.$$

Pour simplifier, on note :

$$c(k,d)^{-1/k} \leq A(u,d)^{-1/k} \left(\frac{1}{\nu_d}\right)^{1/d} \left(\frac{k}{d}\Gamma(k/d)\right)^{-1/k}$$

avec : $A(u,d) = [C_{u,d}^{u/(u+1)}]^{1/k} \left(u^{-u/(u+1)+u^{1/u+1}}\right)$

En utilisant la formule de Stirling on obtient :

$$A(u,d)^{-1/k} \left(\frac{1}{\nu_d}\right)^{1/d} \left(\frac{k}{d}\Gamma(k/d)\right)^{-1/k} \sim A(u,d)^{-1/k} \left(\frac{1}{\nu_d}\right)^{1/d} 2\pi/d k^{1/2-1/d-1/k}$$

$$\sim \left(\frac{1}{\nu_d}\right)^{1/d} 2\pi/d k^{1/2-1/d-1/k}$$

Ceci prouve que :

$$\sum_k c(2k,d)^{1/2k} = +\infty$$

Et qu'ainsi les conditions de Carleman sont vérifiées pour que la loi limite soit définie, de manière unique, par ses moments.

On a ainsi prouvé la convergence en loi de longueurs de liaison, renormalisées, sur le MST.

3.5 Vers un test de connexité sous hypothèse uniforme

3.5.1 Principe

Si X est issu d'un tirage uniforme sur $C = \prod_{i=1}^{d}[0, L_i]$ avec $L_1 \leq L_2 \leq ... \leq L_d$ dans chaque cube de coté L_1 inclus dans C, on compte environ $\frac{L_1^d}{\Pi(L_i)}N$ points. Par indépendance géographique et par homothétie de rapport L_1, soit $\delta(X)$ la variable aléatoire qui donne la taille d'une liaison tirée au hasard sur X. On a

$1/L_1(\frac{L_1^d}{\Pi(L_i)}N)^{1/d}\delta(X)$, quantité qui converge, en loi vers une variable aléatoire de loi notée $\mathcal{L}_d$.

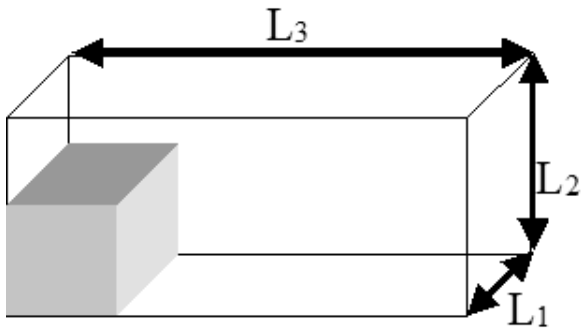


Fig. 3.9:

Dans un premier temps, on ne va pouvoir effectuer des tests que sur des tirages uniformes inclus dans de tels ensembles. A la fin de la partie suivante (Analyse d'une composante connexe), le paramétrage d'une carte de Kohonen permettra de donner des estimations des différentes longueurs L_i dans le cas où l'ensemble des données est homéomorphe à $C = \prod_{i=1}^{d}[0, L_i]$ ce qui inclut d'autres types de classes. Bien sûr l'hypothèse d'uniformité du tirage reste nécessaire.

3.5.2 Quelques résultats

On suppose ici qu'on connaît les Ł$_i$ et on teste la connexité, sous hypothèse uniforme. Les données ont été simulées de la manière suivante : 100 points suivent une loi uniforme sur $[0, 1]^2$ et 100 points sur $[1 + h, 2 + h] \times [0, 1]$ avec $h \in \{2; 1, 5; 1, 3; 1, 2; 1, 1; 1\}$. Idéalement la plus grande rupture de distance intra-classes coincide avec un écart entre les statistiques d'ordre sur les longueurs du MST significatif. Ceci arrive dans les quatre premiers cas. Dans le cinquième cas, la plus grande rupture de distance intra-classes se produit pour une scission en 2 classes, mais l'écart entre $\delta_{(N)}$ et $\delta_{(N-1)}$ n'est pas significatif. Seul l'écart entre $\delta_{(N-4)}$ et $\delta_{(N-5)}$ est significatif. On choisira donc une classification en 5 classes. Enfin le dernier cas est équivalent à un tirage uniforme sur $[0, 2] \times [0, 1]$, il est donc normal qu'aucun écart ne soit significatif. Etant donné qu'on ne dispose que de la convergence de la fonction de répartition en moyenne et qu'on n'a de test que sur l'écart à cette moyenne, on a effectué pour chaque exemple 20 tirages uniformes de 200 points sur $[0, 2 + h] \times [0, 1]$. Les courbes noires représentent les résultats pour ces 20 tests.

Les graphiques se lisent de la manière suivante : pour chaque exemple on observe :

- Diagrammes en batons des distances intra-classes et des différences de distances intra-classes (première ligne, en bleu)
- Diagrammes en batons (rouge) des longueurs, ordonnées, sur le MST (et différences), superposé, en ligne noir, des résultats pour des simulations uniformes
  Remarque : on s'attend a ce que, pour un écart significatif sur les distances du MST on obtienne un écart "significatif" sur les distances intra-classes
- Résultat d'une classification lorsque la rupture des distances sur le MST est jugée significative

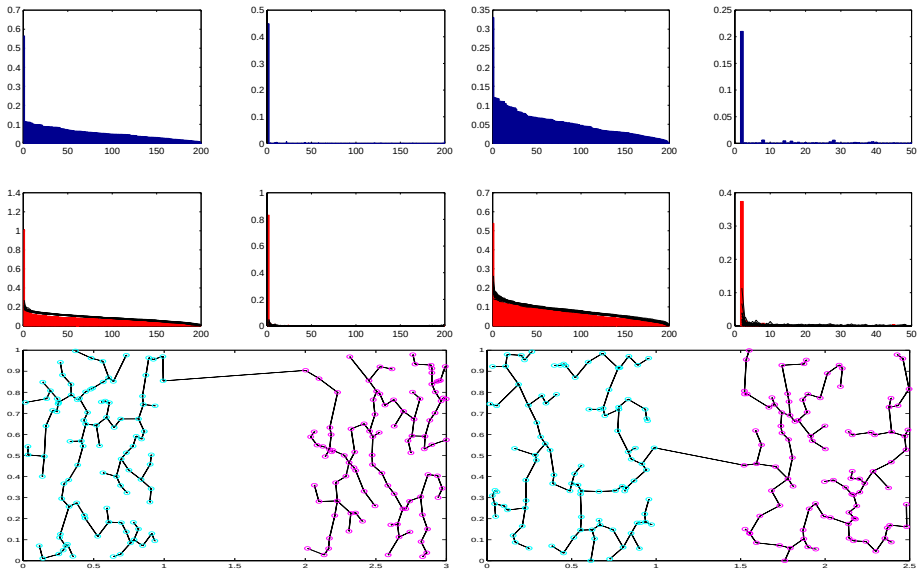


Fig. 3.10: Résultats pour un écart de 2 (à gauche) et un écart de 1.5 (à droite)

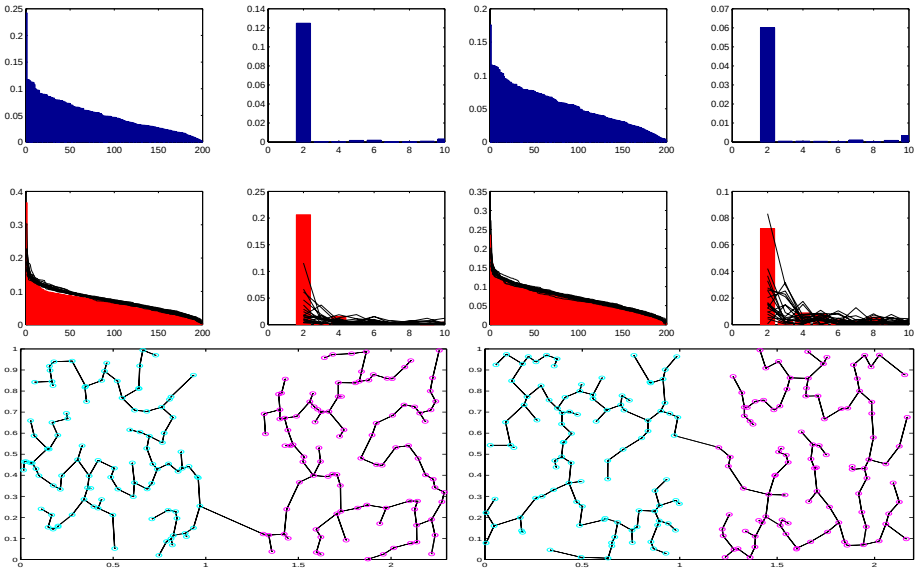


Fig. 3.11: Résultats pour un écart de 1.3 (à gauche) et un écart de 1.2 (à droite)

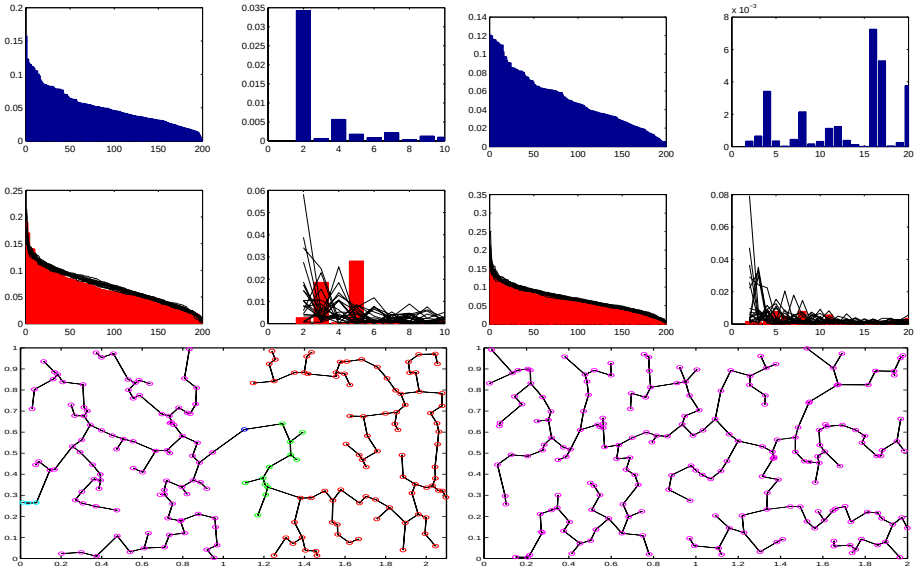


Fig. 3.12: Résultats pour un écart de 1.1 (à gauche) et un écart de 1 (à droite)

3.5.3 Les Limites de la méthode

Il existe deux principales limitations pratiques de la méthode :

- Le fait de ne pas connaître la statistique des écarts de la fonction de répartition empirique (inverse) à la moyenne pour disposer d'un vrai test, et par conséquent d' être obligé d'effectuer des simulations
- Le calcul des L_i dans le cas où les classes ne sont pas linéaires

La première limitation restera en suspend. Mais dans la partie suivante au chapitre 11.3, on trouvera un moyen d'estimer les longueurs L_i dans le cas d'ensembles homéomorphes à $\prod_{i=1}^{d}[0, L_i]$.

4. CLASSIFICATION ET ESTIMATION DE DENSITÉ : ALGORITHME CONJOINT

Dans une première partie, on montre que la classification et l'estimation de densité sont deux problèmes joints. En effet, si on connaît la densité il existe deux méthodes de classification : la méthode du watershed et la méthode du domaine d'attraction des modes. Nous avons déjà exposé rapidement ces méthodes en introduction, mais nous les réexposons ici pour faciliter la lecture. Après avoir rappelé quelques points sur l'estimation de densité par les méthodes à noyaux, on montre ensuite que la connaissance d'une classification peut aider à estimer la densité. Une fois mis en évidence les liens entre les deux problèmes, on construit un algorithme conjoint d'estimation de densité et de classification.

4.1 Classification et estimation de densité : deux problèmes joints

4.1.1 Classification sous hypothèse de densité connue

On fait ici l'hypothèse que les données observées sont issues du tirage d'un mélange de p lois unimodales et, hypothèse supplémentaire, que la densité du mélange fait apparaître p modes. Si la densité f du mélange est connue, deux méthodes de classification ont été proposées dans la littérature : la méthode dite du "WaterShede" et la méthode des domaines d'attraction des modes. Elles ont été introduites dans le chapitre 1, paragraphe 2.

"Water-Shed"

La méthode dite du "Water-Shed" consiste, pour un seuil λ fixé, à regrouper les individus suivant les composantes connexes de $E_\lambda = \{x/f(x) > \lambda\}$

Une telle méthode montre très vite ses limites. Il y a d'une part le choix arbitraire du paramètre λ et, d'autre part, dans des cas

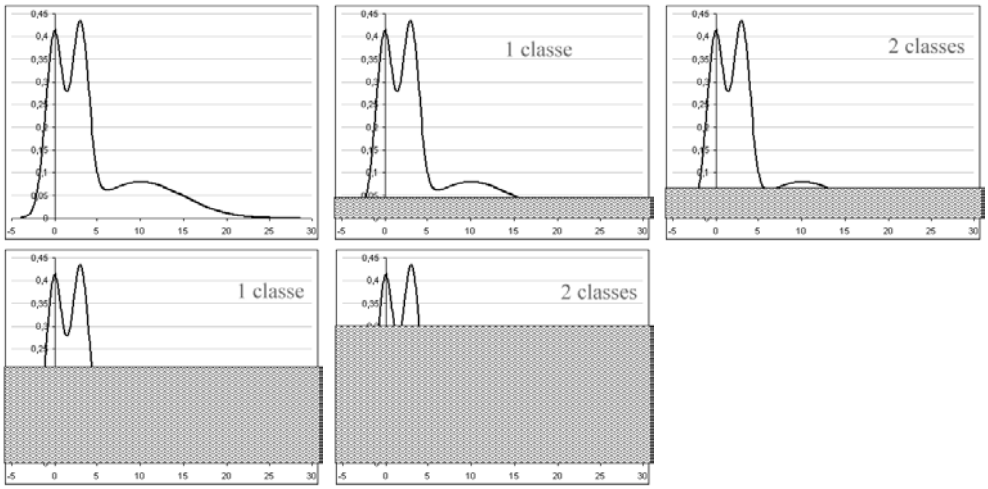


Fig. 4.1: Méthode du Water-Shed : 3 modes, le troisième mode correspond à une dispersion beaucoup plus grande que les deux autres. Pour tout λ, on ne trouve jamais les trois composantes connexes qui sont pourtant présentes dans les données

d'hétérogénéité des dispersions (comme dans l'exemple de la figure 4.1), l'incapacité théorique à trouver conjointement toutes les "vraies classes".

Attraction des modes

Pour répondre aux problèmes posés par la méthode de Water-Shed, Wishart a proposé une nouvelle méthode en 1969 : la classification autour des domaines d'attraction des modes. Dans cette méthode, il y a autant de classes que de modes de la densité et chaque point est affecté à la classe du mode qui peut être relié à ce point par un chemin continu le long duquel la densité est croissante.

On peut présenter les résultats sous la forme d'un dendrograme dont les feuilles correspondent aux modes et les regroupements entre classes aux "points" selles de la densité (en dimension 1 ce sont les minima locaux).

D'une part, cette méthode permet de retrouver les "vraies classes". D'autre part la lecture du dendrogramme associé permet d'obtenir les classes issues du Watershede. Ainsi, d'une certaine manière, cette méthode englobe la précédente, ce qui motive notre choix en faveur de cette méthode, pour la construction de l'algorithme conjoint.

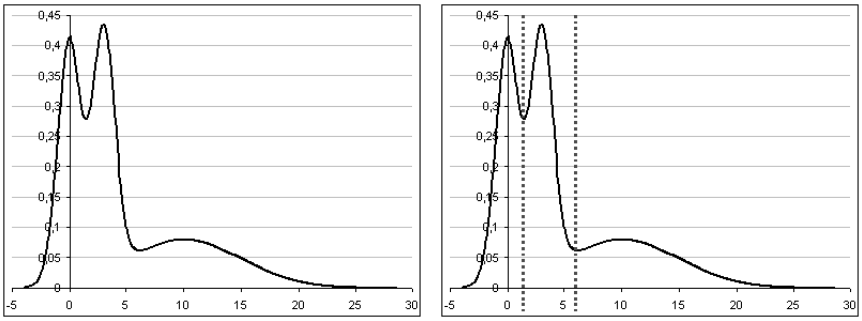


Fig. 4.2: Méthode du domaine d'attraction des modes : pour le même exemple que précédemment, la méthode met bien en évidence les 3 classes.

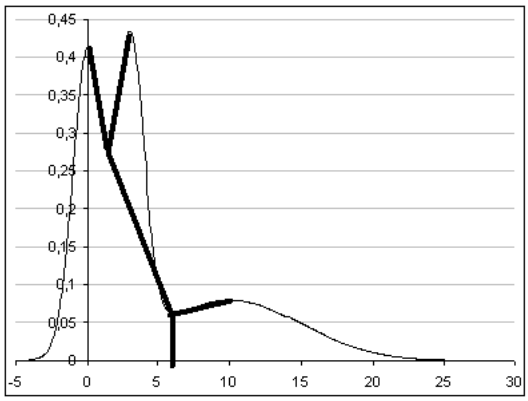


Fig. 4.3: Méthode du domaine d'attraction des modes : dendrogramme associé

4.1.2 Estimation de densité par les méthodes à Noyaux

Dans ce chapitre, pour simplifier les notations, on considére que les données sont unidimensionnelles. Les généralisation des équations au cas multi-dimensionnel sont données en fin de chapitre.

Principe

Les estimations de densité par les méthodes "à noyaux" ont été proposées par Parzen [DEN5] et [DEN6]. Le principe est le suivant :

Soit $K(x, m, h)$ un noyau c'est-à-dire que la fonction $K(., m, h)$ est positive d'intégrale 1, K peut donc être vu comme une densité. Pour que K soit un noyau de Parzen il faut, de plus, pouvoir écrire K sous la forme : $K(x, m, h) = \frac{1}{h} f\left(\frac{x-m}{h}\right)$).

Dans toute la suite on considère un noyau gaussien c'est-à-dire :

$$K(x, m, h) = \frac{1}{\sqrt{2\pi}h} exp\left(-\frac{(x-m)^2}{2h^2}\right)$$

L'estimation de la densité du nuage de points par le noyau K à la taille de fenêtre h est donnée par :

$$f_h(x) = (1/N)\sum_{i=1}^{N} K(x, x_i, h)$$

Le problème majeur des estimations de densité par les méthodes à noyaux réside dans le choix de h, la taille de la fenêtre.

Taille de fenêtre globale

Plusieurs approches pour déterminer la valeur de h ont été étudiées.

Les critères de type $MISE$ (Mean Integrated Squared Error) Localement (en un point x fixé) le biais de $f_h(x)$ comme estimateur de la densité sera d'autant plus petit que h l'est. A contrario la variance sera faible si h est grand.

Les critères de type $MISE$ visent alors à minimiser l'intégrale de la somme du biais (au carré) et de la variance. Ceci mène, au premier ordre, à une formule du type $h_{opt} = h_0 N^{-1/5}$, où h_0 est une constante dépendant de la forme du noyau et aussi de la densité à estimer :

$$h_0 = \left(\frac{||K||_2^2}{||f''||_2^2\, \sigma^4(K)}\right)^{1/5}$$

où $\sigma(K)$ est l'écart type correspondant à la densité K pour une taille de fenêtre 1.

De nombreuses méthodes sont décrites pour estimer h_0, elles reposent toutes sur une estimation de densité qui induit une estimation de f'' qui permet d'estimer une nouvelle taille de fenêtre etc... De telles méthodes sont difficilement adaptables pour la suite où nous avons besoin de considérer des tailles de fenêtre locales.

La maximisation de la pseudo-vraisemblance Une autre approche, initiée par Habbema [DEN4] et Duin [DEN3] pour la recherche de la

taille de la fenêtre h, consiste à maximiser la pseudo-vraisemblance du tirage des données.

L'approche naïve consisterait à maximiser

$$\prod_{i=1}^{N} f_h(x_i),$$

mais alors le maximum est "réalisé" pour $h = 0$ c'est-à-dire pour une somme de Diracs centrées sur chacun des x_i.

Pour pallier ce problème, on utilise la validation croisée, c'est-à-dire qu'on ôte à chaque fois une observation et on calcule la densité en ce point sur la base d'une estimation à noyau reposant sur les points restants.

Pour cela on définit :

$$f_{h,-i}(x) = (1/(N-1)) \sum_{j \neq i} K(x, x_j, h),$$

et

$$LCV(h) = \log(\prod_{i=1}^{N} f_{h,-i}(x_i)).$$

LCV est le logarithme de la pseudo-vraisemblance.

Il apparaît alors que le h^* obtenu par maximisation de LCV converge vers celui qui minimise la distance de Kullback-Leibler :

$$\int \log \frac{f}{f_h}(x) f(x) dx.$$

Tout d'abord remarquons que si les données (x_i) sont toutes différentes :

$$\lim_{h \to 0} LCV(h) = -\infty$$

et

$$\lim_{h \to \infty} LCV(h) = -\infty.$$

Comme $LCV(h)$ est continue sur $\mathbb{R}_+^*$ il existe au moins un maximum local.

La condition du premier ordre en h est alors pour des noyaux gaussiens :

$$h^2 = \frac{1}{N} \sum \frac{\sum_{j \neq i} K(x_i, x_j, h)(x_i - x_j)^2}{\sum_{j \neq i} K(x_i, x_j, h)}$$

Si on considère alors la suite :

$$h^2(n+1) = \frac{1}{N} \sum \frac{\sum_{j \neq i} K(x_i, x_j, h(n))(x_i - x_j)^2}{\sum_{j \neq i} K(x_i, x_j, h(n))}$$

Cette relation s'écrit :

$$h^2(n+1) = h^2(n) + \frac{h^3(n)}{N} \frac{\partial LCV}{\partial h}(h(n)).$$

Ceci garantit la convergence de $h(n)$ vers une taille de fenêtre h^* réalisant un maximum local de LCV car $\frac{\partial LCV}{\partial h}$ est de signe contraire à $h - h^*$ dans un voisinage de $h*$.

Dans cette partie on a considéré la dimension 1. Dans le cas multi-dimensionnel, l'estimation de densité par des noyaux gaussiens s'écrit :

$$\widehat{f}(x) = \frac{1}{N} \sum_{i=1}^{N} K(x, x_i, S^2)$$

où la taille de fenêtre est désormais définie par une matrice de variance-covariance (S^2).

Dans le cas où l'on recherche une matrice de variance-covariance diagonale, l'optimisation de la taille de fenêtre par maximisation de la pseudo-vraisemblance donne pour condition du premier ordre :

$$S^2_{k,k} = \frac{1}{N} \sum \frac{\sum_{j \neq i} K(x_i, x_j, S^2)(x_i^k - x_j^k)^2}{\sum_{j \neq i} K(x_i, x_j, S^2)}$$

et la suite des tailles de fenêtre est donnée par:

$$S^2_{k,k}(n+1) = \frac{1}{N} \sum \frac{\sum_{j \neq i} K(x_i, x_j, S^2)(x_i^k - x_j^k)}{\sum_{j \neq i} K(x_i, x_j, S^2)}$$

Ceci peut aussi s'écrire:

$$S^2_{k,k}(n+1) = {S^2_{k,k}}^2(n) + \frac{S^3_{k,k}(n)}{N} \frac{\partial LCV(S)}{\partial S^2_{k,k}}(S(n))$$

On montre aussi que cette suite converge vers un maximum local de la pseudo-vraisemblance.

Dans le cas où l'on recherche une matrice de variance-covariance quelconque (non diagonale), il est plus facile de travailler sur $M = S^{-1}$. On montre alors que la condition du premier ordre s'écrit :

$$M^2_{k,l} = \left(\frac{1}{N} \sum \frac{\sum_{j \neq i} K(x_i, x_j, S^2)(x_i^k - x_j^k)(x_i^l - x_j^l)}{\sum_{j \neq i} K(x_i, x_j, S^2)} \right)^{-1}$$

Ces calculs ne supposent pas l'existence d'une classification a priori.

4.1.3 Estimation de densité par les méthodes à noyaux sous hypothèse de classification connue

Cas où une seule taille de fenêtre globale est inadaptée

Dans le cas où les données sont issues du tirage d'un mélange de lois avec de fortes disparités de dispersion, ou de grandes différences entre les probabilités d'appartenir à l'une ou l'autre classe, la recherche d'une taille de fenêtre unique pour les noyaux risque fort d'être vouée à l'échec.

Dans l'exemple qui suit, 25 points ont été tirés suivant la loi $N(0, 1)$ et 25 autres points suivant la loi $N(15, 5)$. Les graphiques (figure 5.4) montrent qu'il n'est pas possible d'estimer correctement et simultanément les deux parties de la densité. Le premier graphique présente les résultats pour une taille de fenêtre optimale sur la première partie de la densité, la taille de fenêtre étant trop petite pour la seconde partie. Le deuxième graphique correspond à une taille de fenêtre maximisant la pseudo-vraisemblance. L'estimation y est médiocre partout. Ensuite le troisième graphique illustre le résultat pour une taille de fenêtre adaptée à la seconde partie, mais l'estimation sur la première partie devient alors trop aplatie. Les variations de la pseudo vraisemblance en fonction de la taille de la fenêtre sont représentée au dessous.

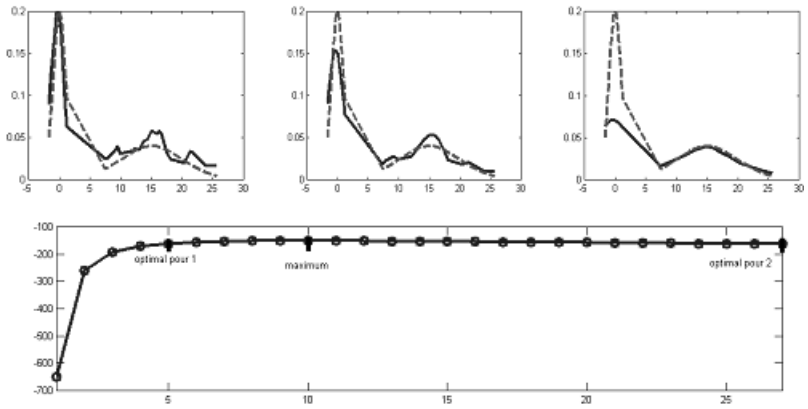


Fig. 4.4: Hétérogénéité des dispersions et estimation de densité

Pour prendre en compte des densités multimodales, plusieurs solutions ont été proposées, on peut citer Abrahamsom [DEN1] qui propose d'estimer la densité par :

$$f(x) = \frac{1}{N} \sum K(x, x_i, h_i)$$

et propose un choix des h_i proportionnel à la racine carrée de $f(x_i)$.

Plus récemment Sain et al [DEN7] proposent une méthode reposant sur le même type de formulation où les h_i sont constants par intervalles. On va dans la suite s'inspirer fortement de cette méthode en considérant les h_i constants par classe.

Si la classification est connue

Supposons que les points sont classés a priori en M classes. On note $cl(i)$ le numéro de la classe du i^{eme} point. Une idée naturelle est de considérer que le nuage est issu d'un mélange de M lois de probabilité dont chaque composante peut être estimée par une méthode à noyaux centrés sur les points de la classe correspondante :

$$f_{h_1,\ldots,h_M}(x) = \sum_{m=1}^{M} p_m \frac{1}{N_m} \sum_{cl(i)=m} K(x, x_i, h_m)$$

où p_m représente la probabilité d'appartenance à la m^{ieme} classe, et N_m est le nombre de points observés dans la m^{ieme} classe.

p_m étant inconnu et estimé par N_m/N, l'expression se simplifie en :

$$f_{h_1,\ldots,h_M}(x) = \frac{1}{N}\sum_i K(x, x_i, h_{cl(i)}).$$

La maximisation de la pseudo-vraisemblance dans le cas de noyaux gaussiens donne alors, comme condition du premier ordre, une équation récursive sur les h_m :

$$h_m^2 = \frac{\sum_{i=1}^N \frac{\sum_{cl(j)=m} K(x_i,x_j,h_m)(x_i-x_j)^2}{\sum_{j\neq i} K(x_i,x_j,h_{cl(j)})}}{\sum_{i=1}^N \frac{\sum_{cl(j)=m} K(x_i,x_j,h_m)}{\sum_{j\neq i} K(x_i,x_j,h_{cl(j)})}}.$$

Dans le cas de noyaux gaussiens, la suite récursive des $h_m(n)$ est donnée à :

$$h_m^2(n+1) = h_m^2(n) + h_m^3(n)\frac{\frac{\partial LCV}{\partial h_m}(h_m(n))}{\sum_{i=1}^N \frac{\sum_{cl(j)=m} K(x_i,x_j,h_m(n))}{\sum_{j\neq i} K(x_i,x_j,h_{cl(j)})}}$$

Cette équivalence justifie à nouveau l'utilisation d'une suite récurrente pour l'estimation des tailles de fenêtre suivant les classes.

4.2 Méthode en dimension 1

4.2.1 Présentation de l'algorithme

Les considérations du chapitre précédent sur le couplage entre les problèmes de classification d'estimation de densité définissent intuitivement un **algorithme conjoint** de recherche des classes et de la densité :

- séparation de l'échantillon en X base d'apprentissage (N_X points) et Y base de test (N_Y points)
- initialisation de l'algorithme
- A chaque étape de l'algorithme :
  - optimisation des tailles de fenêtre par un algorithme itératif sur les données X
  - stockage des résultats

  - recherche des classes par la méthode d'attraction des modes
  - affectation de nouvelles tailles de fenêtre aux classes trouvées
- fin des itérations
- récupération du "meilleur modèle" c'est-à-dire celui qui maximise la vraisemblance sur Y
- on recalcule les tailles de fenêtre pour pouvoir estimer la densité sur la population entière $(X \cup Y)$ en posant $h := h(\frac{N_X}{N_X+N_Y})^{1/5}$
- calcul de la densité sur l'ensemble de la base.
- classement final des points de l'ensemble de la base en fonction des domaines d'attraction des modes de la densité calculée sur l'ensemble des points de la base

Nous allons, dans les paragraphes suivants, spécifier point par point cet algorithme.

Initialisation

Dans le but d'obtenir les résultats les plus "lisses" possibles, on a pris le parti d'effectuer l'algorithme de manière "descendante" c'est-à-dire de partir d'une densité à coup sûr unimodale (une classe) et de scinder celle-ci si cela apparaît nécessaire. Pour garantir l'initialisation sur une classe et une densité unimodale, on écrit l'équation donnant h en fonction des points observés dans le cas d'une seule classe :

$$h^2 = \frac{1}{N} \sum_i \frac{\sum_{j \neq i} K(x_i, x_j, h)(x_i - x_j)^2}{\sum_{j \neq i} K(x_i, x_j, h)}$$

Alors un point de départ correspondant à une densité unimodale est obtenu par une estimation "virtuelle" uniforme. Ce qui donne :

$$h^2 = \frac{1}{N(N-1)} \sum_{j \neq i} (x_i - x_j)^2$$

En résumé, l'initialisation est la suivante :

$\forall i \; cl(i) = 1$
$h_0 = \sqrt{\frac{1}{N(N-1)} \sum_{j \neq i} (x_i - x_j)^2}$

Optimisation des tailles de fenêtre

Version exhaustive On utilise de manière directe la formule de récurrence :

$$h_m^2(t+1) := \frac{\sum_{i=1}^{N} \frac{\sum_{cl(j)=m} K(x_i,x_j,h_m(t))(x_i-x_j)^2}{\sum_{j\neq i} K(x_i,x_j,h_{cl(j)}(t))}}{\sum_{i=1}^{N} \frac{\sum_{cl(j)=m} K(x_i,x_j,h_m(t))}{\sum_{j\neq i} K(x_i,x_j,h_{cl(j)}(t))}}.$$

Le temps de calcul est alors de l'ordre de MN^2 (nombre de classes par le carré du nombre de points)

Approximation Si on fait les approximations suivantes (un peu brutales mathématiquement mais intéressantes en gain de temps de calcul) :

$cl(i) \neq m, cl(j) = m \Rightarrow K(x_i, x_j, h_m) = 0$

L'équation devient :

$$h_m^2(t+1) := \frac{\sum_{i=1}^{N} \frac{\sum_{cl(j)=m} K(x_i,x_j,h_m(t))(x_i-x_j)^2}{\sum_{j\neq i} K(x_i,x_j,h_{cl(j)}(t))}}{N_m}.$$

Recherche des classes par attraction des modes

En dimension 1, cette étape est particulièrement facile, ce qui n'est pas le cas en dimension supérieure. Il suffit d'ordonner les points et de rechercher les minima locaux qui définissent les frontières entre les classes.

Affectation de nouveaux h aux classes

On a pris le parti, toujours dans un soucis de "lissage" maximum de la densité, d'affecter à chaque nouvelle classe la taille maximum de fenêtre des points de la classe.

Récupération du "meilleur modèle"

On considère ici que le meilleur modèle est celui qui a les meilleures capacités de généralisation, c'est-à-dire celui qui maximise :

$\prod(f_{modele}(y_i))$

le modèle étant le couple (classification des points de X, taille de fenêtre en fonction de la classe).

Classification de l'ensemble des points de la base

Il faut désormais donner une classification de l'ensemble des points de la base. On n'a classé que les points de la partie "base d'apprentissage". Pour la partie "test" (Y), on affectera à chaque y_i la classe du x_j le plus proche de y_i.

Densité finale

Pour finir, on va estimer la densité en centrant les noyaux sur chacun des points de la base (apprentissage+test). La théorie de l'estimation de densité par noyaux nous dit alors qu'il faut pondérer les tailles de fenêtres, pour prendre en compte l'augmentation du nombre de points, par $(\frac{N_X}{N_X+N_Y})^{1/5}$.

4.2.2 Résultats

Les graphiques suivants présentent des résultats numériques en estimation de densité et en classification. Dans un premier temps, les exemples considérés sont issus de simulations (3 échantillons différentes pour chacune des 4 lois considérées) avec 50 points en base de test et 50 points en base d'apprentissage).

Les graphiques se lisent verticalement dans l'ordre :

- vraisemblance sur la base d'apprentissage
- vraisemblance sur la base de test
- densité estimée sur la seule base d'apprentissage et densité théorique
- densité estimée finale et densité théorique

Dans un second temps, on a testé la méthode sur des bases de données réelles classiquement utilisées pour tester les méthodes d'estimation de densité. La base des "buffalo snowfalls" et "Old Faithfull". Dans les deux cas, il existe des données "répétées" dans la base, ce qui est incompatible avec la méthode exposée. Pour pallier ce problème, il a été remarqué que les données étaient arrondies à 10^-1 dans le cas des "buffalo snowfalls" et 10^-2 pour "Old Faithfull". On a ainsi ajouté un bruit uniforme respectivement sur $[-5\times10^{-2}, 5\times10^{-2}]$ et sur $[-5\times10^{-3}, 5\times10^{-3}]$. Les effectifs

des bases d'apprentissage et des bases de test sont respectivement de 40 et 27 (buffalo) et 57 et 50 (Old Faithful). Enfin dans chacun des cas, on a effectué 10 essais dont on présentera ici le "pire", le "moyen", et le "meilleur" (au sens des capacités de généralisation observées sur le maximum de pseudo vraisemblance sur Y).

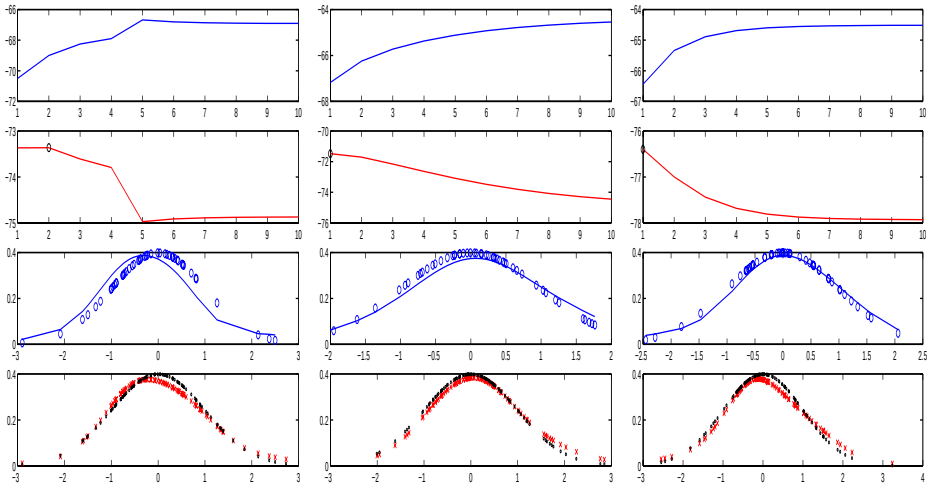


Fig. 4.5: Résultat d'estimation pour des tirages gaussiens : première ligne la pseudo vraisemblance lors de l'apprentissage, sur la seconde la vraisemblance sur la base test, sur la toisième ligne densité estimée (plein) et théorique (ronds) sur la base test, sur la dernière ligne densité estimée (rouge) et théorique (noire) sur la base de validation

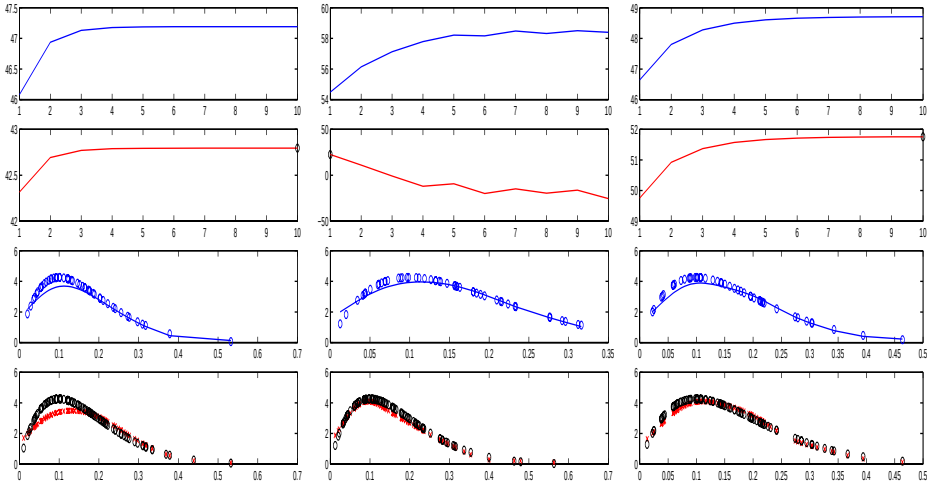


Fig. 4.6: Résultats pour des lois béta $(2, 10)$

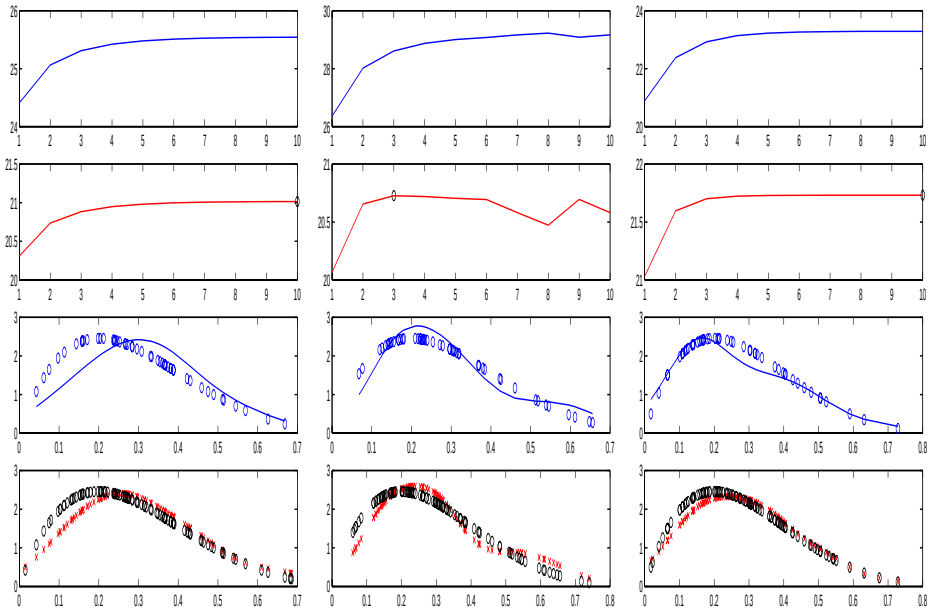


Fig. 4.7: Résultats pour des lois béta $(2, 5)$

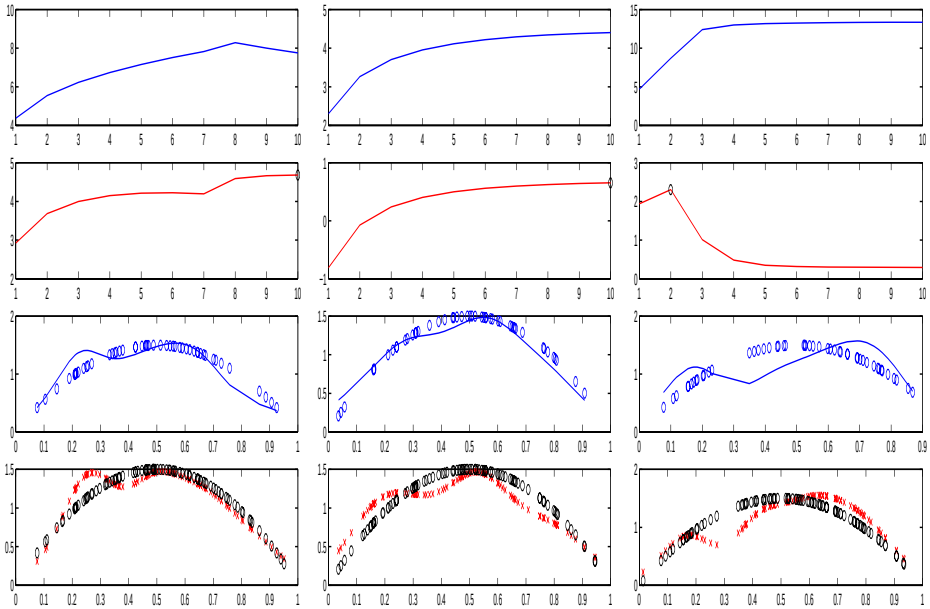


Fig. 4.8: Résultats pour des lois béta $(2, 2)$

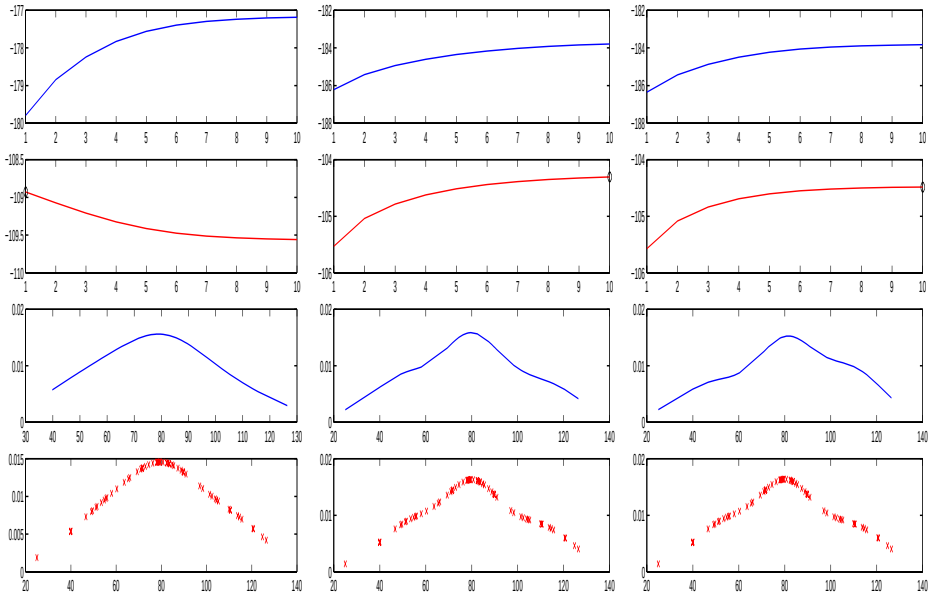


Fig. 4.9: Buffalo SnowFall la "pire", la "moyenne" et la "meilleure" sur 10 essais

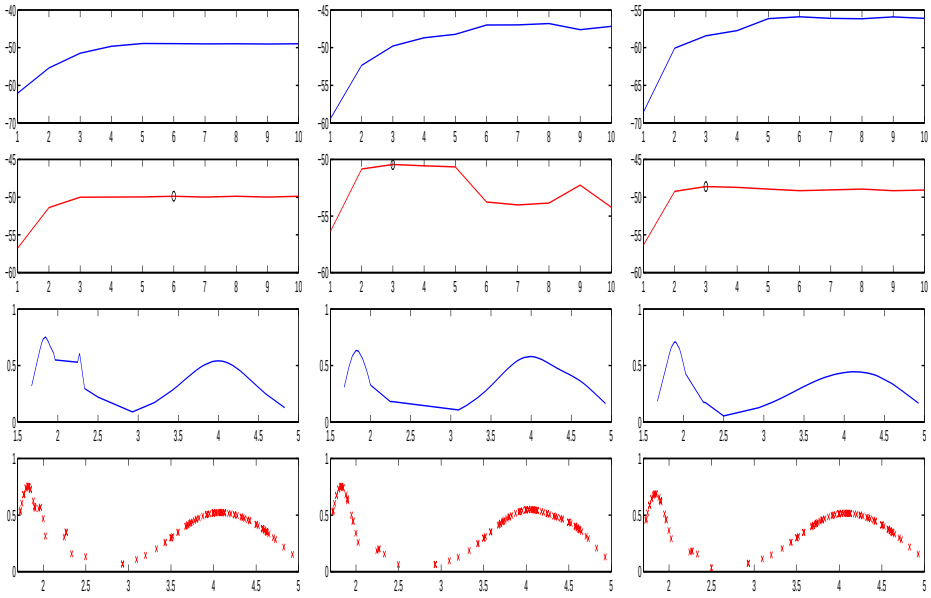


Fig. 4.10: Old Faithfull data la "pire", la "moyenne" et la "meilleure" sur 10 essais

4.3 Méthode en dimension quelconque

4.3.1 Ce qui change

Les équations d'adaptation des tailles de fenêtre

Lorsque l'échantillon des données $\vec{X}_i = (X_i^1, ..., X_i^d)$ est dans $\mathbb{R}^d$, les tailles de fenêtres sont désormais des matrices de variance-covariance à rechercher dans l'ensemble des matrices symétriques de dimension d, $S \in S(d)$.

Les équations de recherche du maximum de la pseudo-vraisemblance deviennent celles énoncées au chapitre 4.1.2.

Deux possibilités sont classiquement envisagées :

- recherche de matrices de variance-covariance diagonales (adaptation des seuls coefficients diagonaux) et :

$$S_{k,k}^2(n+1) = \frac{1}{N} \sum \frac{\sum_{j \neq i} K(x_i, x_j, S^2)(x_i^k - x_j^k)}{\sum_{j \neq i} K(x_i, x_j, S^2)}$$

- recherche de matrice de variance covariance quelconques (adaptation de tous les coefficients) et

$$((S^{-1})_{k,l}^2(n+1) = \left(\frac{1}{N} \sum \frac{\sum_{j \neq i} K(x_i, x_j, S^2)(x_i^k - x_j^k)(x_i^l - x_j^l)}{\sum_{j \neq i} K(x_i, x_j, S^2)} \right)^{-1}$$

Si la seconde méthode donne de meilleurs résultats en terme d'estimation de densité, elle est moins stable que la première, notamment dans le cas où les données sont en fait dans un sous espace de $\mathbb{R}^d$ de dimension inférieure (on aura des problèmes de matrices non inversibles).

Un autre désavantage de cette méthode réside dans la difficulté d'affectation des nouvelles matrices de variance-covariance après reclassification des données.

La recherche de la classification associée

Là aussi le passage en dimension multiple complique les choses. On ne peut plus se contenter de localiser les minima locaux pour effectuer la classification.

La méthode "exhaustive" consistant à rechercher pour chaque point le maximum local obtenu par montée de gradient sur la densité donne certes le résultat optimal mais n'est guère envisageable au regard du temps de calcul d'une telle méthode.

Une méthode qui permet d'approcher ce résultat consiste en la construction d'un graphe orienté qui lie chaque point $\vec{X_i}$ au point le plus proche $\vec{X_j}$ vérifiant $f(\vec{X_j}) \geq f(\vec{X_i})$. Un tel graphe relie tous les points au maximum global de la densité. Pour lier chaque point uniquement au maximum local de son domaine d'attraction, il faut supprimer du graphe précédent les liaisons entre les maxima locaux et les points qui se trouvent dans un domaine d'attraction d'un maximum local plus élevé.

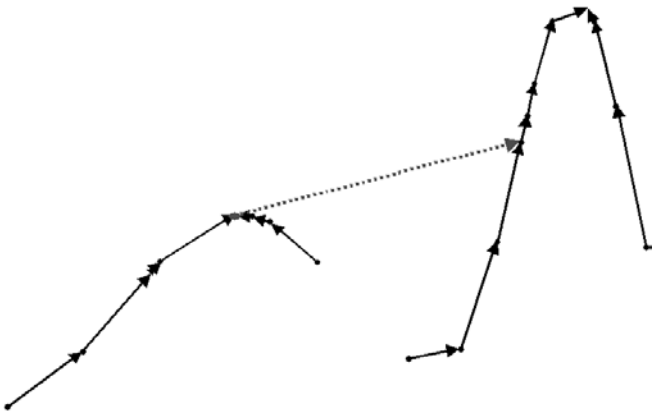


Fig. 4.11: Exemple de graphe orienté, en pointillé la liaison à supprimer

Pour cela on teste, pour chaque liaison l'existence d'un minimum local de densité lorsqu'on parcourt la liaison. Le gain de temps est non négligeable (la recherche d'un minimum pouvant se faire par dichotomie). On obtient un temps de calcul "raisonnable" mais néanmoins long.

On propose ici une méthode moins rigoureuse mais très rapide qui repose sur la construction d'un graphe local de la manière suivante :

- Calcul du MST
- En chaque point x, on ajoute toutes les liaisons de taille inférieure à la plus grande taille de liaison partant de x. Le graphe obtenu est noté G

- **Il faut alors supprimer les liaisons de G sur lesquel il existe un minimum local**
- Etant donné que le graphe est "local", on se contente de supprimer les liaisons entre x et y si $f((x+y)/2) < min(f(x), f(y))$

Enfin la classification des points s'effectue suivant les composantes connexes de G.

Affectation de nouvelles tailles de fenêtre après classification

Dans le cas de la recherche de matrice de variance-covariance diagonales, on procède, comme dans le cadre unidimensionnel, à l'affectation des plus grandes variances afin d'obtenir, à chaque étape, une densité la plus "lisse" possible, et ceci, direction par direction :

$$\forall i, j \ S_i(j,j)(t+1) = max_{Cl(k)(t+1)=i}\{S_{Cl(k)(t)}(j,j)(t)\}$$

Densité finale

En dimension d, la pondération des tailles de fenêtres, pour prendre en compte l'ensemble de tous les points, devient $(\frac{N_X}{N_X+N_Y})^{1/(d+4)}$.

4.3.2 Résultats

On présente ici (figure 4.12) le résultat de cette méthode (algorithme conjoint) pour la segmentation des Iris de Fischer en dimension 2 (deux premiers axes d'une ACP sur les 150 Iris caractérisées par 4 variables de tailles sur leurs pétales). La réduction en dimension 2 a été effectuée, d'une part pour pouvoir avoir un graphique lisible, d'autre part car il s'avère, numériquement, que pour de trop grandes dimensions les résultats ne sont pas stables (ici, par exemple les deux premiers axes expliquent 98 de l'inertie totale, lorsqu'on prend en compte l'ensemble des quatre axes les matrices de variance-covariance s'approchent de matrices non inversibles).

4.4 Conclusion

La méthode proposée est très efficace du point de vue de l'estimation de densité en dimension 1 ou 2 lorsque les modes ne sont pas trop "plats",

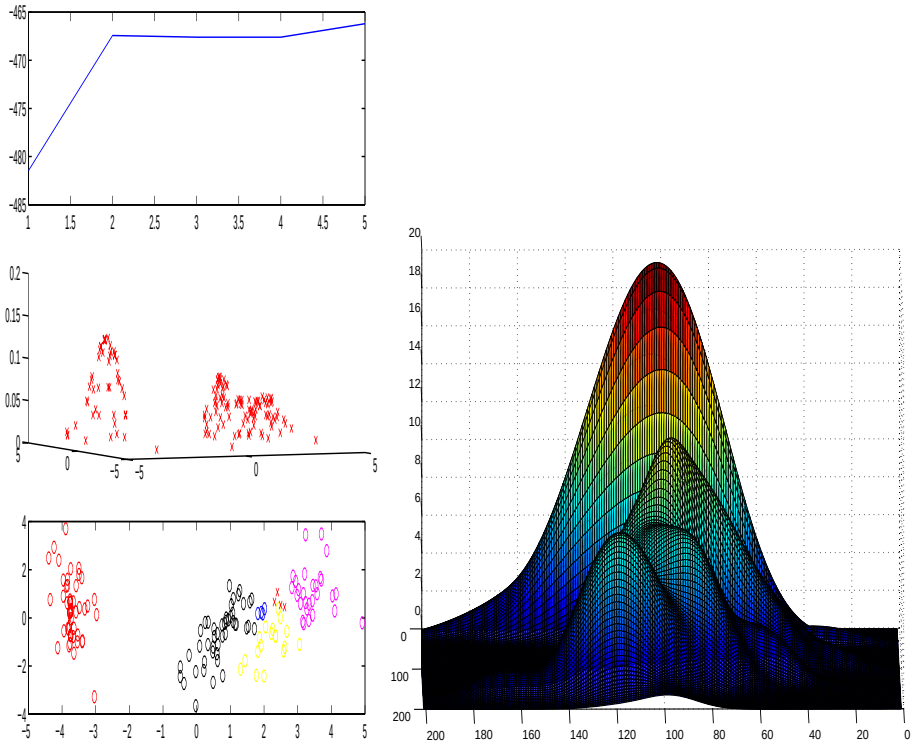


Fig. 4.12: Iris de Fisher, évolution de la vraisemblance, densité estimée sur la base test et nappe correspondante sur l'ensemble des données et résultat de la classification en 5 classes

mais les choses sont moins évidentes en dimension supérieure : D'une part il existe un problème de "visualisation" des résultats. D'autre part le nombre de points nécessaire à une "bonne" estimation de la densité en dimension d quelconque est très grand (il faut estimer d paramètres lorsqu'on estime la densité avec des matrices de variance-covariance diagonales et $d(d-1)/2$ paramètres si on ne fait pas d'hypothèses sur la forme de la matrice).

Pour ce qui est de l'estimation de densité, on peut considérer que les résultats sont corrects dans le cas où les modes sont "pointus" et bien différenciés. Mais ceci n'est pas vraiment compatible avec les objectifs de classification en composantes connexes. Dans le chapitre suivant nous construisons à l'aide des deux approches précédentes (classification hiérarchique avec distance du minimum et algorithme conjoint) une méthodologie sophistiquée de classification en composantes connexes

prenant en compte la potentielle hétérogénéité de dispersions au sein des classes.

5. STRATÉGIE DE CLASSIFICATION FINALE

On propose ici une stratégie finale de classification qui repose sur les deux méthodes de classifications en composantes connexes exposées précédemment (classification hiérarchique avec la distance du minimum et approche mixte avec la densité). L'objectif est ici de pouvoir repérer et séparer des composantes connexes hétérogènes en dispersion. Pour cela on va calculer le MST qui est lié à la classification hiérarchique de la manière suivante : Si on supprime les K plus grandes liaisons du MST on obtient la classification des points suivant les composantes connexes du graphe en $K+1$ classes (qui correspond à celle obtenue par classification hiérarchique avec la distance du minimum). On estimera la densité (et la classification associée) en travaillant sur les longueurs des liaisons du MST. Suivant les résultats d'estimation de densité on pourra "voir" si les différentes classes sont homogènes ou hétérogènes en dispersion. Et choisir, en fonction une méthode de classification adaptée au données.

5.1 Résumé des avantages et inconvénients des deux méthodes proposées

5.1.1 La classification à l'aide de la densité

On a de très bons résultats en dimension 1 si les modes ne sont pas trop "aplatis". En dimension 2 les résultats restent corrects pour des observations en nombre "raisonnable", en revanche dès la dimension 3, le nombre de paramètres du modèle est tel qu'il faut trop d'observations pour que la méthode soit envisageable (d'autant plus que le temps de calcul est relativement long). De plus la condition de "modes pas trop aplatis" est incompatible avec des composantes connexes non convexes.

5.1.2 La classification hiérarchique

La classification hiérarchique par la distance du minimum, avec un calcul de la distance intra-classe fondé sur la connexité, donne des résultats

"parfaits" si, pour tout couple de classes C_i et C_j respectivement δ_i et δ_j connexes, on a : $d_{min}(C_i, C_j) > max(\delta_i, \delta_j)$.

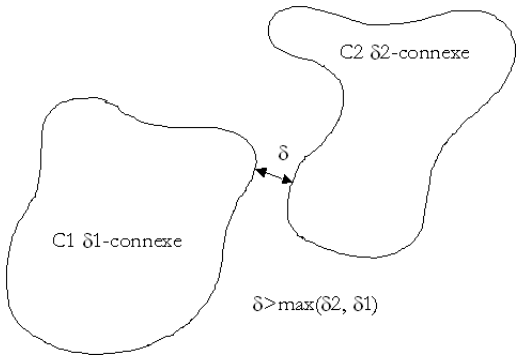


Fig. 5.1: Cas où la classification hiérarchique sépare les classes

Les problèmes peuvent survenir, soit lorsque deux classes de seuils de connexité équivalents sont trop proches (mais, dans ce cas la significativité des deux classes est elle même discutable), soit dans le cas d'hétérogénéité des dispersions au sein des classes. Le cas limite d'hétérogénéité ayant lieu lorsque, pour séparer deux classes, il faut isoler tous les points d'une des deux classes (une configuration correspondante est illustrée dans la figure suivante):

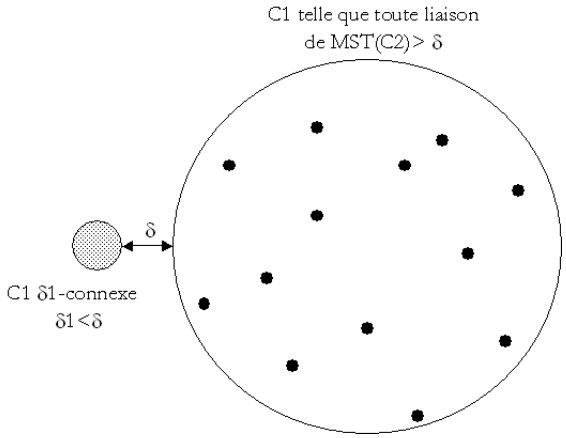


Fig. 5.2: Cas où la classification hiérarchique échoue

5.1.3 La conjugaison des deux méthodes

Dans les cas d'hétérogénéité de la dispersion au sein des classes, le MST qui correspondra "à peu près" à l'union des MST sur chacune des classes ainsi que des liaisons entre classes aura, lui aussi, des longueurs de liaisons très hétérogènes. Si l'hétérogénéité de dispersion au sein des classes est suffisamment élevée alors la densité des liaisons sur le MST sera multimodale et on s'aidera de la classification utilisant la densité des liaisons du MST pour classer les points. Dans les cas d'homogénéité des dispersions au sein des classes, on obtiendra une densité unimodale qui sera un indicateur de choix de la classification de type hiérarchique. Il se peut bien sûr que l'on obtienne des densités unimodales alors que la dispersion au sein des classes est suffisamment forte pour que la classification hiérarchique échoue. On arrivera néanmoins à effectuer des classifications "correctes" dans un plus grand nombre de cas à l'aide d'une conjugaison des deux méthodes.

5.2 Stratégie de classification finale

5.2.1 Classification des points en fonction d'une classification sur la longueur des liaisons du MST

Supposons que la densité estimée sur les longueurs de liaison du MST ait donné lieu à une densité multimodale et donc à une classification L en k classes (des liaisons). On va alors classer les points comme il suit :

Pour i de 1 à $N-1$

- On supprime la liaison δ_i du MST (ce qui nous donne un graphe G_i)
- On classe les points suivant les composantes connexes de G_i (on obtient deux classes). Notons C_i la classification associée
- On affecte à chaque liaison la classe d'une de ses extrémités et on obtient une nouvelle classification C'_i sur les liaisons. (Remarque : les deux extrémités d'une liaison sont forcément dans la même classe par propriété des MST)
- On construit le tableau croisé entre C'_i et L dont on calcule le $\chi^2(i)$

On garde la classification C'_{i_0} avec $i_0 = argmax(\chi^2(i))$

Si on a deux classes hétérogènes Cl_1 et Cl_2 et une seule liaison du MST qui les connecte, la méthodologie proposée va retrouver les classes de manière relativement bonne.

En effet on observe que les liaisons de la classe la moins dispersée (supposons C_1) se trouvent toute dans la classe du plus faible mode, alors que les liaisons de la classe la plus dispersée (C_2) sont à la fois dans les deux classes. Supprimer la liaison entre C_1 et C_2 réalise alors le maximum du $\chi^2(i)$, si cette liaison n'est pas dans la classe 1. Si la liaison entre les deux classes n'est pas dans la classe 1 ,il se peut que sa suppression maximise le $\chi 2$, sinon on ôtera une liaison relativement proche de cette dernière.

Si il existe plusieurs liaisons du MST qui connectent des points de C_1 et C_2, alors la méthode proposée ne retrouvera jamais la "bonne" classification. En effet en ne supprimant qu'une liaison, une des deux classes obtenues par suppression de la liaison mélangera C_1 et C_2. Si on tentait de construire un algorithme similaire en supprimant plusieurs liaisons d'un coup, on arriverait à des temps de calcul bien trop longs pour un résultat garanti "mauvais". En effet dans l'hypothèse où il existe $n > 1$ liaisons connectant C_1 a C_2, il faudrait les supprimer toutes pour isoler une des classes ce qui nous donnerait une classification en $n + 1$ classes (et non en 2).

Dans ce cas on préférera, à l'issue de chaque étape, éventuellement segmenter de nouveau les classes obtenues.

5.2.2 Présentation de l'algorithme final

Pour un nuage de points X, on effectue la classification hiérarchique par la distance du minimum et le diagramme des distances intra-classes associé. On effectue aussi une estimation de densité sur les longueurs de liaisons sur le MST. Quatre cas sont alors envisageables :

- (1) : On n'observe pas de rupture de distance intra-classes et la densité des longueurs de liaison est unimodale. On décide alors que le nuage de points est connexe

- (2) : On observe une rupture de distance intra-classes grande pour un nombre de classes "raisonnable" k (par raisonnable on entend "pas trop élevé") et la densité des longueurs des liaisons est unimodale. On choisit alors une classification hiérarchique en k classes

- (3) : On n'observe pas de rupture de distance intra-classes et la densité des longueurs des liaisons est multimodale. On choisit alors de classer les points en fonction des classes des liaisons et on réitère le processus sur les deux classes obtenues

- (4) : On observe une rupture de distance intra-classes et la densité des longueurs des liaisons est multimodale. Les deux méthodes peuvent donner des résultats intéressants et sont à tester. Étant donné que la classification hiérarchique est plus simple et qu'elle n'impose pas de réitération, il peut sembler plus simple de choisir cette méthode.

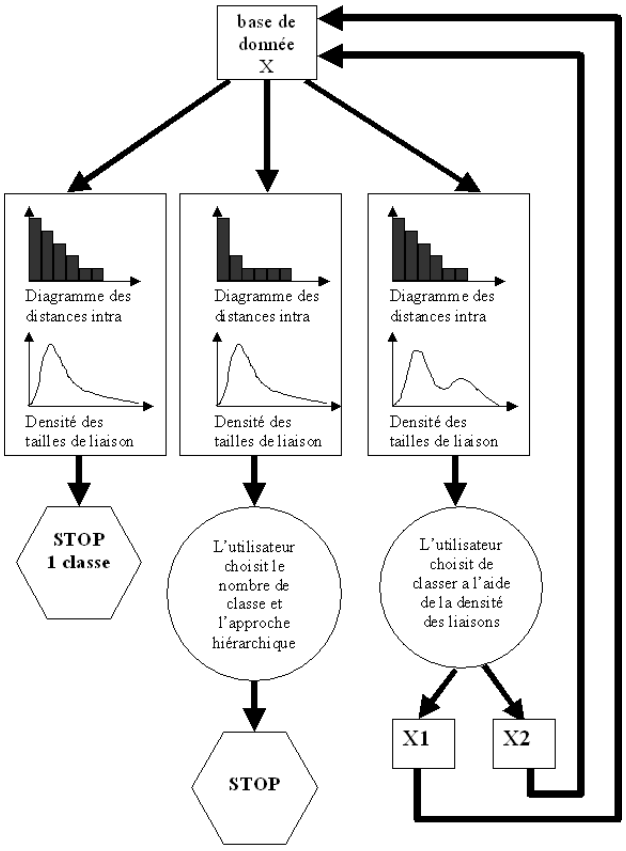


Fig. 5.3: Méthodologie finale de classification

5.3 Résultats

5.3.1 Un exemple où les approches hiérarchique seule et mixte avec densité fonctionnent

On a simulé des données comme il suit : 50 points suivent une loi normale centrée et de variance $(1/5)I_2$ et 150 points sont situés sur une couronne avec un angle réparti uniformément sur $[0, 2\pi]$ et un rayon réparti uniformément sur $[1.2; 4]$.

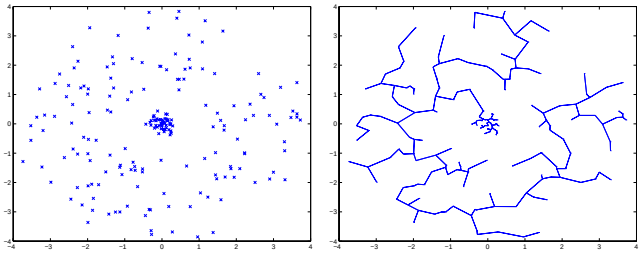


Fig. 5.4: Base de donnée et MST associé

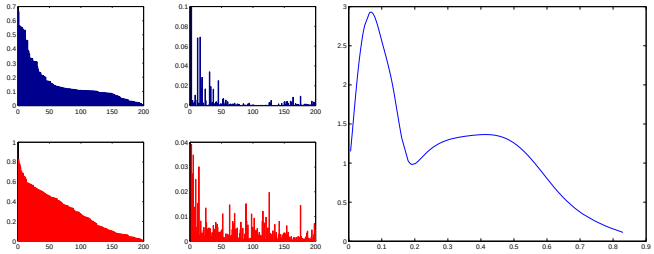


Fig. 5.5: Diagrame des distances intra-classes et estimation de densité sur les tailles des liaisons

Pour la classification hiérarchique seule, une classification en deux classes s'impose et il existe une hétérogénéité de dispersion au sein des classes.

La classification mixte (c'est-à-dire par coupure du MST en fonction des classes de liaison) donne des résultats assez satisfaisants : aucune des classes n'est à scinder et seuls deux points sont mal classés.

L'approche hiérarchique seule retrouve parfaitement les classes.

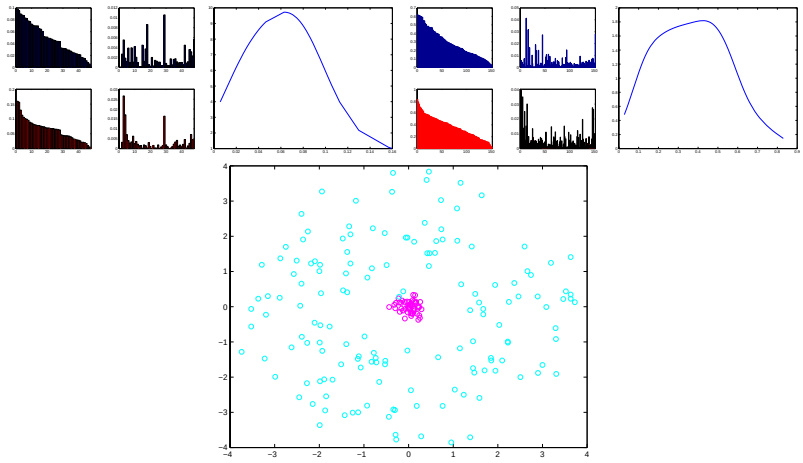


Fig. 5.6: Classification par densité : pas de scission des deux classes obtenues et classification associée

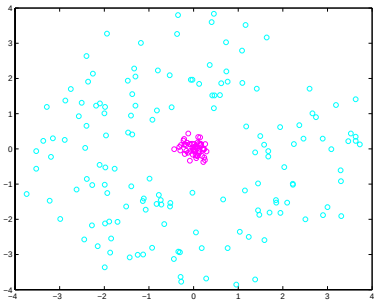


Fig. 5.7: Résultats pour l'approche hiérarchique seule

5.3.2 *Un exemple où l'approche mixte retrouve correctement les classes alors que la hiérarchie seule échoue*

On a simulé des données comme il suit : 50 points suivent une loi normale centrée et de variance $(1/5)I_2$ et 150 points sont situées sur un cercle avec un angle répartit uniformément sur $[0, 2\pi]$ et un rayon répartit uniformément sur $[1.1; 4]$. C'est-à-dire que, par rapport à l'exemple précédent, on a simultanément rapproché la seconde classe de la première et élargit sa dispersion.

On ne voit pas vraiment de scission pertinente pour la hiérarchie seule, l'estimation de densité donne lieu à une fonction multimodale, on adopte donc la deuxième méthode :

On doit scinder la première classe par approche hiérarchique et conser-

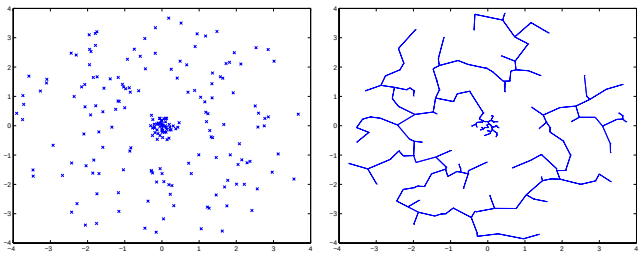


Fig. 5.8: Base de donnée et MST associé

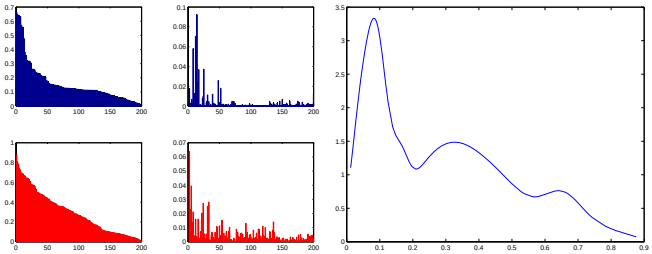


Fig. 5.9: Diagrame des distances intra-classes et estimation de densité sur les tailles de liaison

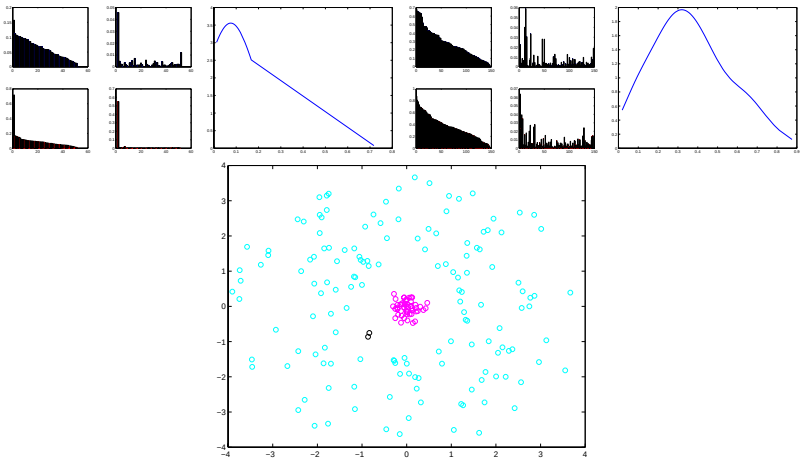


Fig. 5.10: Classification par densité : scission de la première classe par approche hiérarchique et conservation de la seconde; classification associée

ver la seconde, les résultats sont alors relativement bons puisque seuls deux points de la seconde classe sont isolés dans une troisième classe.

A titre indicatif on présente la classification par hiérarchie seule pour

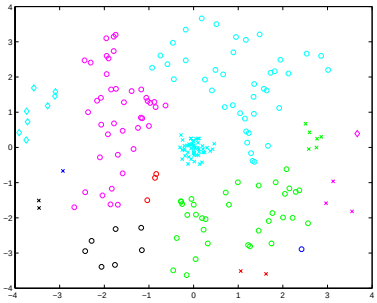


Fig. 5.11: Résultats pour l'approche hiérarchique seule

le nombre de classes maximisant la rupture de distance intra-classes, c'est-à-dire 14.

5.3.3 *Un exemple où il faut ré-itérer l'approche mixte*

On a encore rapproché la première classe de la seconde en accroissant sa dispersion, ceci en simulant des données comme il suit : 50 points sont issus d'une loi normale centrée et de variance $(1/5)I_2$ et 150 points sont situés sur un cercle avec un angle réparti uniformément sur $[0, 2\pi]$ et un rayon réparti uniformément sur $[1; 4]$.

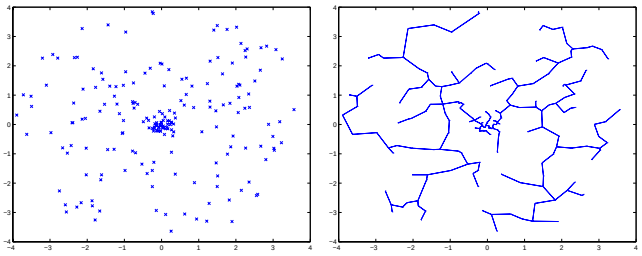


Fig. 5.12: Base de donnée et MST associé

La visualisation du MST nous montre que deux liaisons joignent les classes 1 et 2 ce qui nous indique que l'on va devoir ré-itérer la classification. Bien sûr, dans le cas plus réaliste de la classification non supervisée, les classes sont inconnues et la dimension trop grande pour permettre une telle visualisation, le graphe du MST n'est présenté ici que pour illustrer les phénomènes qui peuvent se produire.

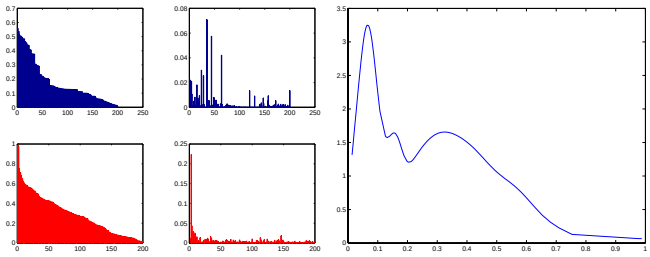


Fig. 5.13: Diagramme des distances intra-classes et estimation de densité sur les tailles de liaison.

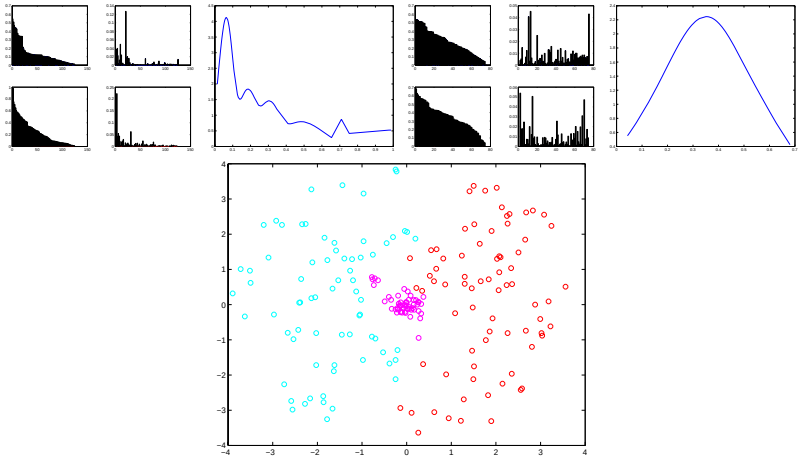


Fig. 5.14: Classification par densité : scission de la première classe par approche mixte hiérarchie/densité et conservation de la seconde; classification associée.

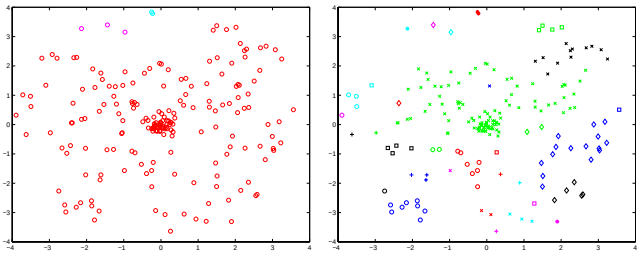


Fig. 5.15: Résultats d'une classification hiérarchique seule pour 3 et 34 classes (maximum de rupture de distance intra-classe)

5.3.4 Un exemple où l'approche hiérarchique seule donne les résultats attendus

Le meilleur moyen de simuler des données sans hétérogénéité de taille des liaisons entre les différentes classes est de simuler deux classes de même

nature. C'est pourquoi on va travailler ici sur des tirages gaussiens de même variance et avec autant de points dans une classe que dans l'autre. On a donc tiré 50 points suivant une loi normale centrée de matrice de covariance identité et 50 points suivant une loi normale centrée en $(2.5; 2.5)$ et de matrice de covariance identité.

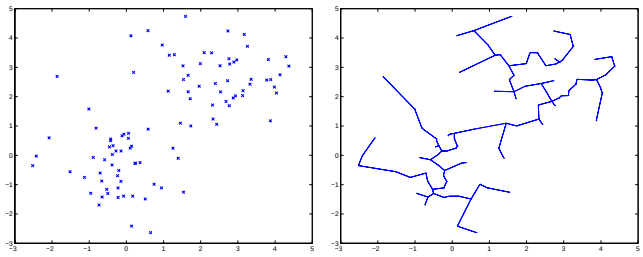


Fig. 5.16: Base de donnée et MST associé

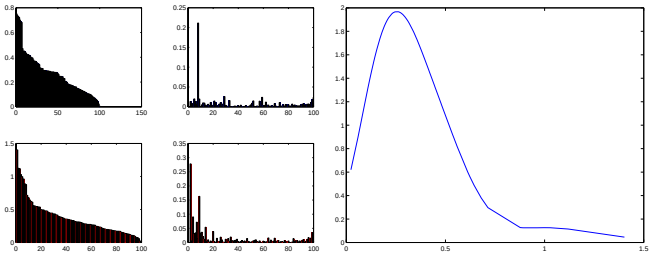


Fig. 5.17: Diagramme des distances intra-classes et estimation de densité sur les tailles de liaison

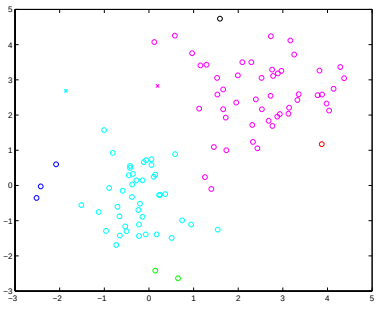


Fig. 5.18: Classification en 8 classes par le choix de la méthode hiérarchique

Une remarque s'impose ici, la densité estimée sur les liaisons est multimodale mais on a choisi l'approche hiérarchique pour deux raisons : d'une

part une scission en 8 groupes semble significative pour la classification hiérarchique, d'autre part, dans le cas où la classification hiérarchique fonctionne, on va supprimer toutes les liaisons de longueur supérieure à un certain seuil, liaisons qui peuvent très bien former un mode de la densité. C'est exactement le cas ici, comme on peut le voir sur le graphique de la densité si on ajoute les points permettant d'observer combien de liaisons forment le second mode:

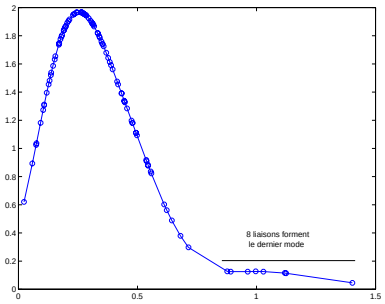


Fig. 5.19: $6-8$ liaisons forment le second mode, la classification hiérarchique en 8 groupes les supprime, on choisit donc cette méthode.

5.4 Conclusion et perspectives

La méthodologie proposée donne de très bons résultats dans les cas d'hétérogénéité de dispersion si il n'existe qu'une liaison sur le MST joignant les classes à séparer. Dans le cas où il en existe plusieurs, il faudrait pouvoir les supprimer en une seule étape. Le principal problème est qu'alors, il faudrait tester les classification en composantes connexes du MST auquel on a supprimé un nombre non fixe de liaison ce qui devient un problème NP complet. Une première perspective serait alors de développer un algorithme "rapide" convergeant vers une solution "correct" à ce problème.

D'un tout autre point de vue, on note que l'étude de la densité estimée sur les longueurs de liaisons permet, de manière relativement efficace, d'observer si il y a, ou non, hétérogénéité de dispersion. Dans le cas d'hétérogénéité de dispersion les méthodes de classification classiques (en classes convexes) atteignent aussi leur limites. On pourrait ainsi adapter ces méthodes au cas d'hétérogénéité de dispersion et, on peut supposer que dans le cas où les classes sont convexes de telles méthodes donneront

de meilleurs résultats que celle proposée. Alors, comme énoncé en fin de chapitre sur la classification hiérarchique avec la distance du minimum, on pourra mettre en place la stratégie suivante :

- Classification sous hypothèse de connexité
- Test de séparabilité linéaire en essayant de retrouver les classes par une analyse discriminante (linéaire)
- Si les classes sont séparables par des hyper-plans, on les sépare avec des algorithmes "classiques".

6. CONCLUSION ET PERSPECTIVES

On a construit une méthode de classification qui repose très fortement sur la notion de connexité qui donne des résultats encourageants sur les classes observées et sur la détermination du nombre de classes. Reste encore à finaliser en mettant en oeuvre les tests de convexité (séparabilité par des hyperplans) proposés dans les perspectives des chapitres 2 et 5. D'un point de vue général, la méthode semble néanmoins légèrement moins robuste que les nouvelles méthodes de classification spectrales surtout dans le cas où deux classes sont liées par plus d'une liaison du MST. Mais ces dernières méthodes (classification spectrale) nécessitent un grand nombre de paramètres. On pourra, par la suite, tenter d'utiliser conjointement les deux méthodes afin d'obtenir, une idée sur les paramètres (notamment le nombre de classes) et des classes robustes.

D'un point de vue théorique il reste à démontrer, pour la partie "vers un test de connexité" un lemme pour avoir une démonstration de la convergence des histogrammes. Il faudrait aussi pouvoir, connaître la vitesse de convergence de la "variance" des fonctions de répartitions empiriques des longueurs pour pouvoir avoir un test "numérique" et plus seulement une indication graphique.

Dans la pratique, on reparle du test de connexité dans la partie suivante où l'on estimera les "longueurs" d'un ensemble dans les différentes "directions" si l'ensemble est non linéaire (longueurs qui sont nécessaires à la mise en place du test).

Enfin il serait intéressant d'étudier les propriétés de la méthode jointe de classification et d'estimation de densité en terme d'estimation de densité, notamment de comparer les résultats obtenus avec d'autres méthodes d'estimation de densité et, surtout, de voir si on peut adapter le couplage densité/classification aux méthodes d'estimations de densité par des ondelettes.

Part II

ANALYSE D'UNE COMPOSANTE CONNEXE : RECHERCHE DE DIMENSION ET PROJECTION

INTRODUCTION

Dans toute cette partie, on va supposer que l'on dispose d'un nuage de points X constitué d'une seule classe connexe que l'on désire analyser. Comme on l'a déjà souligné dans la partie précédente, une notion aussi simple que celle du barycentre n'est plus pertinente si on ne se place que sous la seule hypothèse de connexité. Ainsi, par exemple, dans le cas de la normalisation, le fait de normaliser les données en fonction des écarts au barycentre devient inadéquat sous notre hypothèse.

L'idée principale de l'analyse de composantes connexes réside dans l'utilisation de la distance curviligne à la place des distances usuelles (euclidiennes, L^k...). Une telle distance est de plus en plus fréquemment utilisée dans les méthodes d'analyse des données non linéaires (on peut citer par exemple ISOMAP ou "curvilinear distance analysis"). Cependant, on montrera dans une première partie que la normalisation à effectuer en préliminaire au calcul de la distance curviligne est très importante, et on exposera un algorithme de normalisation qui permet de tenir compte des "non-linéarités" potentielles.

Une fois les données normalisées et la distance curviligne calculée, on définira un indicateur central correspondant au barycentre pour cette nouvelle distance, ce qui constituera un premier pas vers l'analyse des classes.

Ensuite on exposera les méthodes d'estimation de la dimension intrinsèque, méthodes qui nous intéressent pour deux raisons : d'une part, en elle-même la dimension est un indicateur primordial pour comprendre et analyser une classe, et elle permet de paramétrer correctement les méthodes de projection non linéaire des données ; d'autre part, dans l'optique "modélisation" (que l'on précisera en conclusion et perspectives) pour qu'un modèle $Y = f(X)$ avec f continue existe, il faut, d'une part que X et $G(X, f) = \{(x, f(x)), x \in X\}$ soient connexes mais aussi que la dimension de $G(X, f)$ soit celle de X.

Enfin on présentera des méthodes non linéaires de projection des données et de réduction de dimension. On s'intéressera plus particulièrement aux cartes de Kohonen pour lesquelles on construira un indicateur de respect de la topologie qui permettra de valider a posteriori

la projection et d'avoir une idée plus précise de la dimension intrinsèque des données.

1. NORMALISATION DES DONNÉES

1.1 Introduction

Sous la seule hypothèse de connexité, qui permet d'avoir des formes d'ensembles fortement non linéairement séparables, les nouvelles méthodes d'analyse des données non linéaires telles qu'ISOMAP ou "curvilinear distance analysis" ([CRV1], [CRV2], [CRV3], [CRV4]), qui reposent toutes deux sur la notion de distance curviligne (versus la distance euclidienne dans le cas de l'analyse des données linéaire), semblent ouvrir des perspectives intéressantes sur la compréhension des données.

Dans la pratique, la distance géodésique (ou curviligne) entre deux points dans un ensemble E est définie comme la plus petite des longueurs des chemins continus liant ces deux points. La figure suivante montre en quoi cette distance est plus pertinente que la distance euclidienne dans le cas de l'analyse des données non-linéaires.

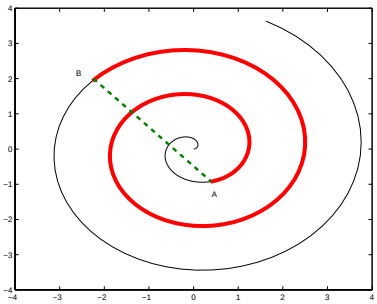


Fig. 1.1: Distances curviligne et euclidienne dans le cas de points situés sur une spirale

Dans le cadre théorique "parfait" où l'on possède une infinité (dense) de points, il n'y a pas de problème de normalisation des données, les distances curvilignes entre les points seront ordonnées de la même manière quelque soient les échelles choisies suivant les différents axes.

Pour des données "réelles", c'est-à-dire des observations discrètes et

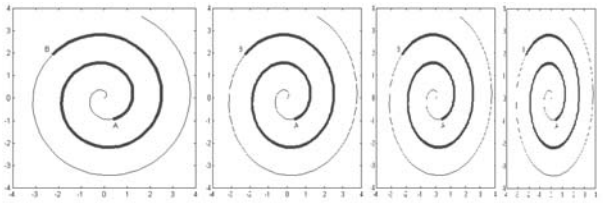


Fig. 1.2: Cas d'ensembles denses

en nombre fini de réalisations d'une variable aléatoire, le calcul de la distance curviligne repose sur le choix d'un graphe connexe (de type MST, $K-$plus proches voisins...) liant les points, et sur l'algorithme de Dijkstra. Dans ce cas le graphe de liaison sera extrêmement sensible aux échelles choisies pour les différents axes.

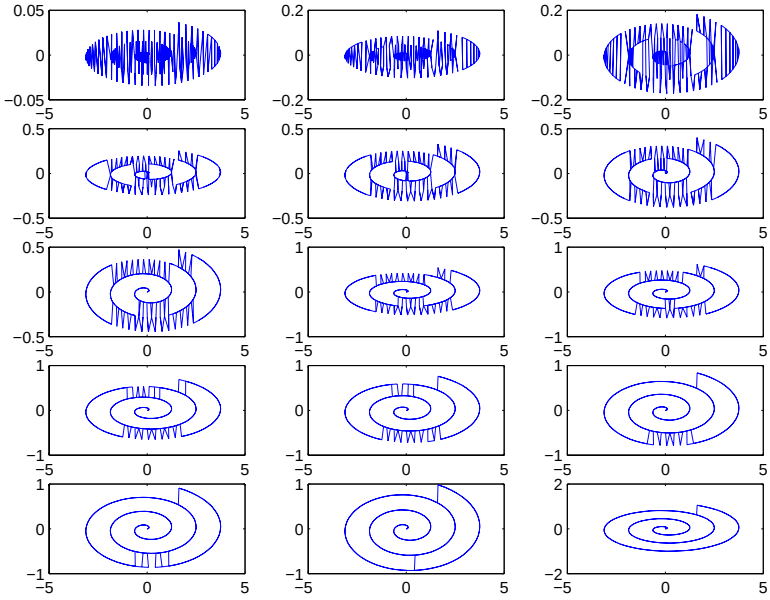


Fig. 1.3: Graphe des $2-$plus proches voisins sur une spirale pour différentes échelles de l'axe vertical

Cet exemple montre l'intérêt d'une normalisation préliminaire au traitement des données, mais n'est pas particulièrement bien choisi dans le sens où la normalisation "classique" (division par l'écart type) donne un "bon" graphe. Par "bon" graphe on entend, dans ce chapitre, un graphe rendant correctement compte de l'organisation des données.

Un exemple beaucoup plus intéressant est donné par des ensembles

"sinusoïdaux" (ici X est tiré uniformément sur $[0,1]$ et $Y = sin(\omega X)$), comme on le constate sur le graphique suivant représentant le graphe des $3-$plus proches voisins dans le cas d'une normalisation "classique" et pour différentes valeurs de ω.

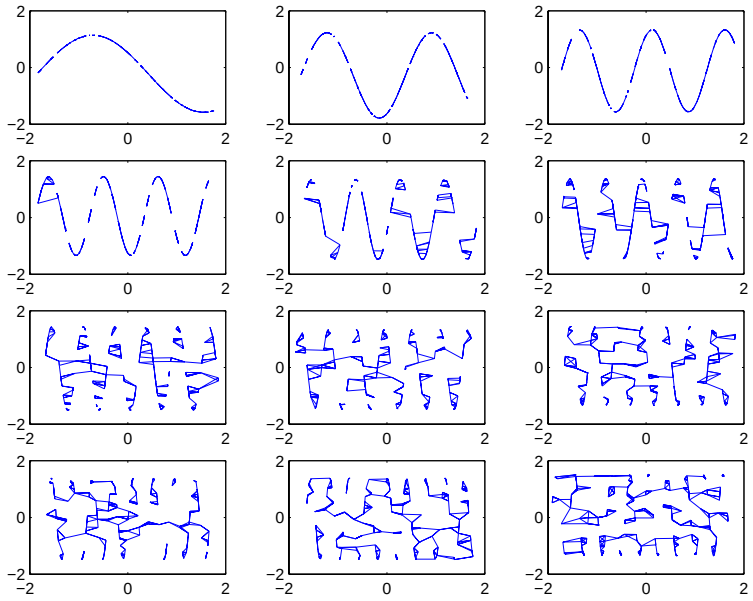


Fig. 1.4: Graphes des trois plus proches voisins pour une normalisation classique et : $Y = sin(\omega X)$, $\omega \in \{5, 10, ...55, 60\}$.

Dans la suite on présente des résultats pour des ensembles sinusoïdaux, car ils sont caractéristiques de formes géométriques pour lesquelles les méthodes de normalisation "classiques" échouent.

1.2 Normalisation simple

1.2.1 Normalisation par des graphes

Méthodes Euclidienne et non Euclidienne

Observons tout d'abord que les normalisations reposant sur des critères de dispersion globale (distance moyenne au barycentre) seront mises en défaut dans le cas de données fortement non linéaires telles que les exemples sinusoïdaux précédents (et ce quelque soit la distance utilisée dans le critère de dispersion retenu). Dans les cas fortement non-linéaires,

on doit utiliser un critère local qui permet de définir, pour chaque point, un voisinage "acceptable" de ce point. Pour cela le principe mis en oeuvre par l'algorithme présenté ci-après consiste à rendre les liaisons du graphe isotropes (c'est-à-dire ici, de "longueur" équivalente dans toutes les directions).

Algorithme : la version déterministe

Poids des axes

Pour un graphe G (assimilé à sa matrice représentative : $G(i,j) = 1$ si x_i et x_j sont liés et $G(i,j) = 0$ sinon), on définit le poids w_j de l'axe j par :

soit $\vec{u_j}$ un vecteur directeur normalisé du j^{eme} axe

$$w_j = (1/N(G)) \sum_{i \neq j, G(i,j)=1} |x_i \vec{x}_k . \vec{u_j}|$$

avec

$$N(G) = \sum_{i \neq j, G(i,j)=1} 1$$

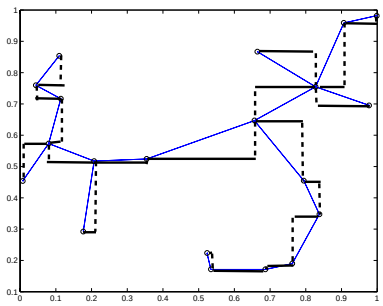


Fig. 1.5: Exemple de contribution de chaque axe (en maigre : MST, en plein contribution horizontale, en pointillé contribution verticale) pour 20 points uniformément distribués sur $[0,1]$

Algorithme

Dans un premier temps, on choisit une structure de graphe représentant les points ($k-$ plus proches voisins, MST...), puis à chaque pas de

l'algorithme on itére :

- (1) calcul de : G
- (2) $\forall j$ calcul de w_j
- (3) $\forall j$ on divise la j^{eme} composante de x_i par w_j

Remarques :

- Cet algorithme tend à rendre le poids de chaque axe égal à 1. Une telle solution n'existe pas forcément (et n'est pas forcément unique), le critère d'arrêt doit donc contenir une notion de proximité à la solution "tous les poids égaux à 1" et un nombre d'itérations maximal

- Le fait que le graphe change après chaque pondération des axes impose une approche algorithmique.

Résultats

Voici quelques résultats pour l'algorithme ci-dessus. Pour chaque ensemble choisi, on présente le MST pour des données normalisées de manière "classique" (initialisation de l'algorithme) et le MST en sortie de l'algorithme.

Le graphe choisi pour la normalisation est le MST.

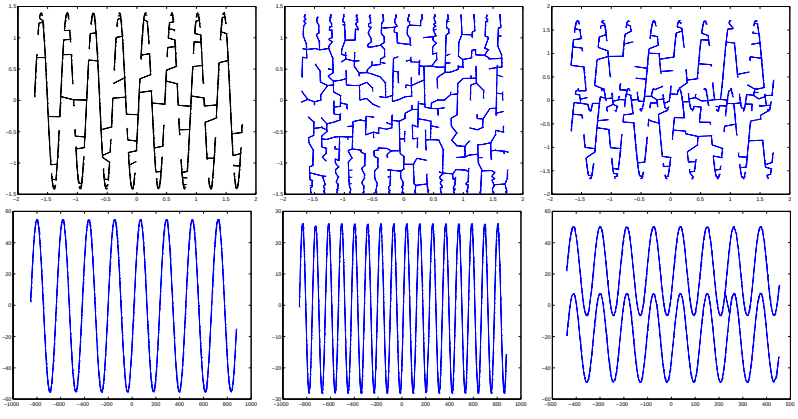


Fig. 1.6: Trois exemples d'organisation des points lors de la normalisation. Sur la première ligne le MST calculé sur les données centrées réduites de manière "classique" ; sur la seconde les données MST-normalisées

Version stochastique de l'algorithme

On propose ici une version stochastique de l'algorithme précédent qui a deux avantages. D'une part, d'un point de vue purement pratique le calcul des graphes est long (en $O(N^2)$ au mieux suivant les graphes considérés) et le travail sur des sous-ensembles procurera ainsi un gain de temps considérable.

L'algorithme ne change pas fondamentalement par rapport à celui exposé dans la section précédente. Les paramètres d'entrées deviennent : $NbIt$ le nombre d'itérations et $N' < N$ la taille des sous-ensembles sur lesquels on va travailler. L'algorithme devient naturellement :

tant que $it <= NbIt$

- (1) tirage de $N' < N$ qui forment X' un sous-ensemble de X
- (2) calcul de $G(X')$
- (3) $\forall j$ calcul w_j sur $G(X')$
- (4)$\forall j$ division de la j^{eme} composante de x_i par w_j (pour l'ensemble des données et plus seulement le sous-ensemble sélectionné)

Comme la taille des liaisons pour le sous-ensemble X' tiré aléatoirement est inférieure a la taille des liaisons sur X l'algorithme ne tend pas vers une solution où le poids de chaque axe pour $G(X)$ vaut 1, mais vers une égalité du poids de tous les axes.

On présente ici le résultat pour $N = 1000$ points sur une sinusoïde en dimension 3 (X et Y tirés uniformément dans $[0, 1]$ et $Z = sin(40X)$). Les paramètres choisis pour faire tourner l'algorithme sont : $N' = 500$, $NbIt = 50$ et on a choisi une structure de graphe aux $8-$plus proches voisins. Comme précédemment le premier graphique représente le graphe pour une normalisation "classique" et le second à l'issue de la normalisation.

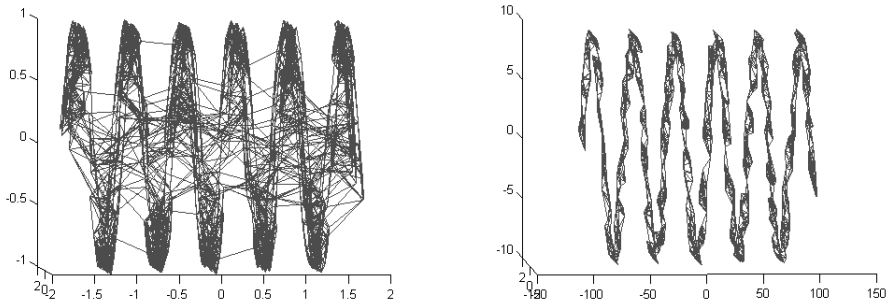


Fig. 1.7: Resultats pour une sinusoïde 3-D. A gauche la normalisation "classique" a droite la normalisation sur les graphes dans une version stochastique

1.2.2 Impact sur la distance curviligne

On voudrait savoir ici dans quelle mesure l'algorithme permet de répondre au problème présenté en introduction, c'est-à-dire retrouver la distance curviligne entre les points. Pour cela on va observer l'évolution de la liaison entre la distance curviligne estimée et la distance curviligne théorique dans des exemples où l'on connaît la distance curviligne théorique.

Pour cela on a simulé 100 observations en dimension 2 de la manière suivante : X^1 est une réalisation d'un tirage uniforme sur $[0, 1]$ et $X^2 = sin(\omega X^1)$. On a testé différentes valeurs de ω : $\omega \in \{20, 25, ..., 45\}$. Pour illustrer la nécessité de la normalisation, les graphiques initiaux correspondent à la normalisation classique (division par l'écart-type) et on a lancé la normalisation fondée sur le MST dans sa version stochastique avec des tirages de 75 points à chaque étape. L'algorithme a été itéré 8 fois et le nuage de points distance curviligne théorique et distance curviligne "approchée" est présenté pour les étapes 1 : (normalisation classique), 4 (au milieu des itérations) et 8 (à la fin des itérations).

Plusieurs remarques s'imposent ici :

Tout d'abord la distance curviligne théorique ne dépendant que de la distance sur X^1 on a choisi cet indicateur.

Les résultats sont bons pour des valeurs de ω inférieures à 40 sans qu'on puisse savoir si le mauvais résultat (pour $\omega = 40$) est du à une forme de saturation ou à un trop petit nombre d'itérations.

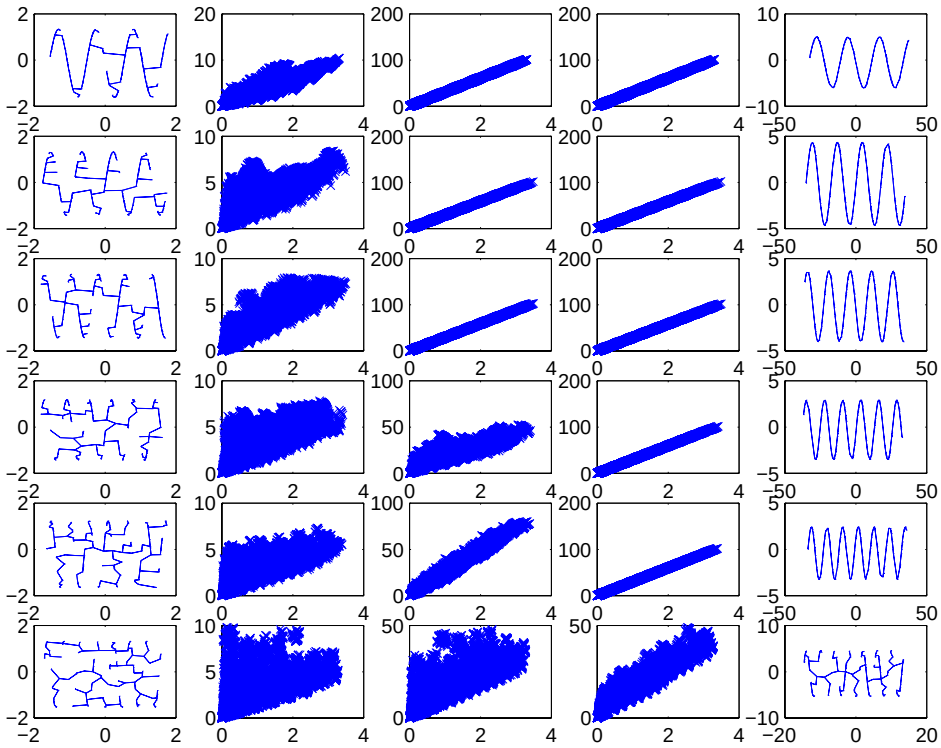


Fig. 1.8: Correlation entre la distance curviligne théorique et "approchée" au cours de l'algorithme de normalisation sur le MST.

1.3 Normalisation et recherche des axes principaux

1.3.1 Motivation

Lorsque l'ensemble des variables n'est pas, comme dans les cas précédents, constitué d'un sous-ensemble de variables indépendantes et d'un sous-ensemble de fonctions de ces variables ($X^1, ..., X^k$ variables indépendantes et $X^{k+1} = f_{k+1}(X^1, ..., X^k), ..., X^p = f_p(X^1, ..., X^k)$) mais que de telles variables ont subi, par exemple, une rotation, l'algorithme de normalisation "simple" peut être mis en défaut. Ceci est totalement équivalent, en analyse des données linéaires, à normaliser des données avant d'avoir effectué une ACP. Dans ce cas, il existe néanmoins une correction simple à ajouter à l'algorithme qui permet de résoudre le problème.

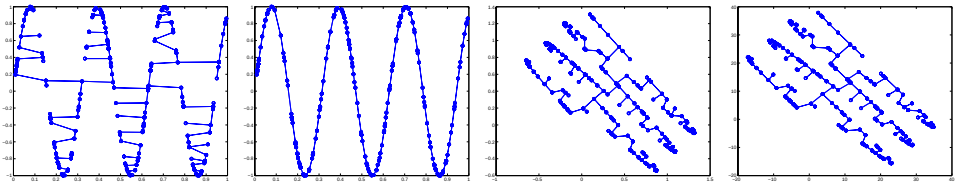


Fig. 1.9: Effet de la normalisation sur un ensemble de type sinusoïdal et sur le même ensemble après rotation de $-\pi/4$: dans le premier cas la structure est retrouvée alors que dans le second cas rien ne change.

1.3.2 Algorithme

L'algorithme précédente modifié pour prendre en compte les rotations éventuelles des données est le suivant. Il consiste à itérer la séquence, toujours après avoir choisi un type de graphe représentant les voisinages (les opérations ayant changé par rapport à l'algorithme précédent sont indiquées en gras.):

- Calcul du graphe
- Stockage des liaisons (comptées dans les deux sens)
- **Calcul d'une ACP sur les vecteurs liaisons**
- **Application de l'ACP aux données (comme on applique une rotation le graphe des données avant et après reste identique)**
- Calcul des poids w des nouveaux axes (après ACP)
- Division des nouvelles données par le poids de leur axe

Remarque : Comme les liaisons sont stockées dans les deux sens (c'est-à-dire que si A et B sont liés sur le graphe, les deux vecteurs $\vec{AB}$ et $\vec{BA}$ sont stockés) l'ACP est bien calculée sur un nuage recentré.

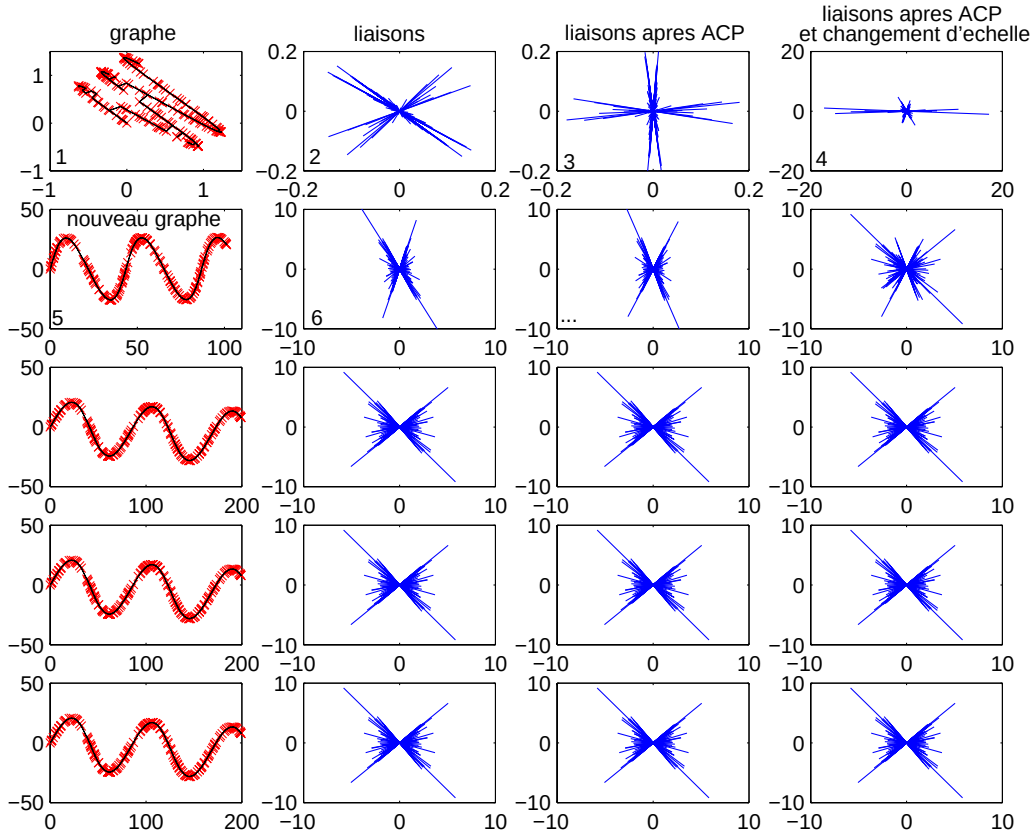


Fig. 1.10: Illustration pas à pas de l'algorithme.

La proximité avec le poids unitaire de chaque axe n'est plus le seul critère d'arrêt : il faut aussi que les rotations dues aux ACP successives convergent. Ainsi, à la place d'un critère d'arrêt "simple" comparant le poids des axes à 1, on va préférer itérer un certain nombre de fois les calculs, et observer l'évolution des différents critères de convergence au cours du temps : poids des axes et angle maximal de rotation (modulo $\pi/2$).

Remarque : en préliminaire, il est nécessaire d'effectuer une *ACP* sur les données afin de ne conserver que des axes dans lesquels les données sont représentées (inertie non nulle) afin de ne pas diviser par 0 dans l'étape 6 de l'algorithme.

1.3.3 Résultats

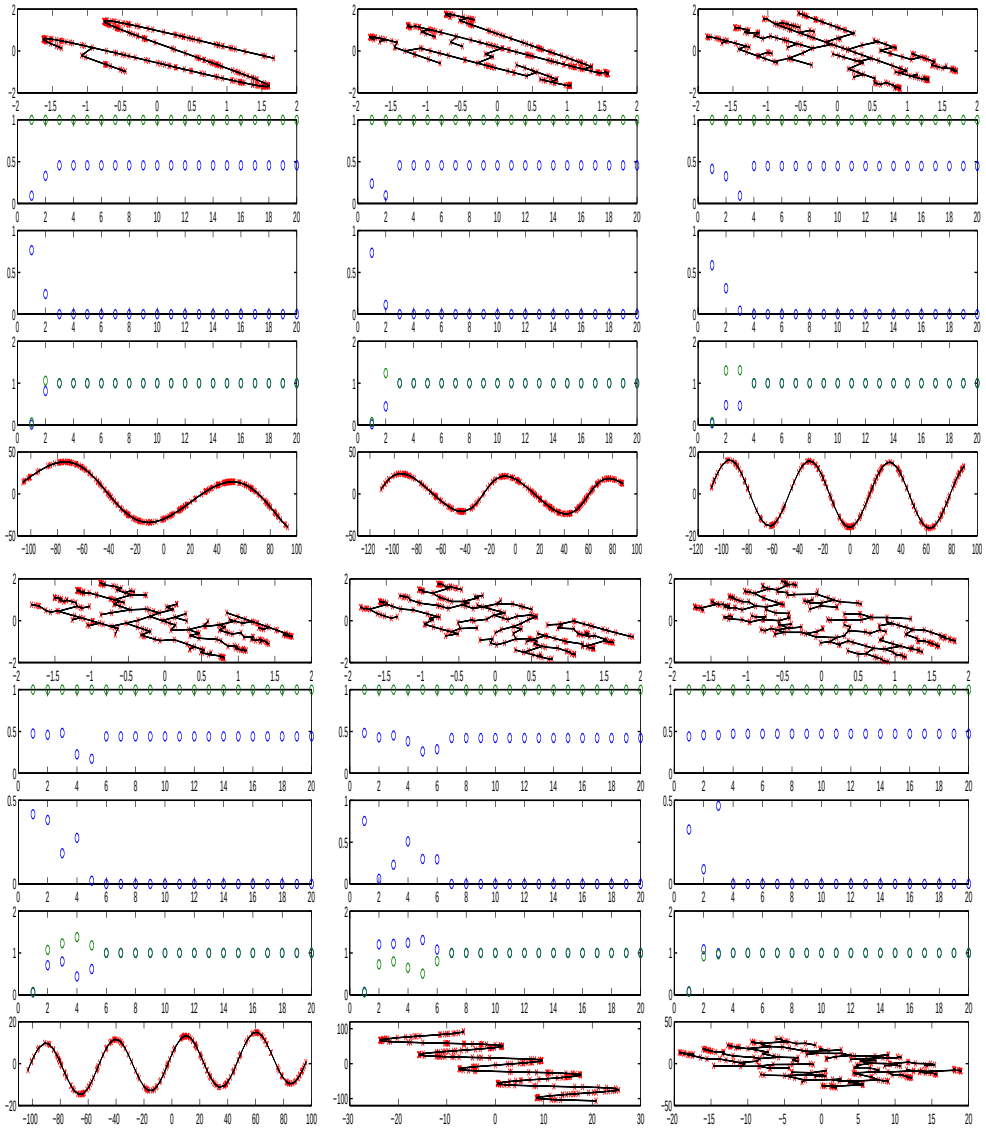


Fig. 1.11: Résultats de la normalisation avec rotation pour 200 points tirés uniforméments sur des sinosoïdes de fréquence 10 à 25 : pour chaque exemple (qu'on lit verticalement) le premier graphe représente le MST sur les données normalisées de manière "classique", le deuxième graphe représente la contribution (cumulée) de chaque axe a l'inertie (dans l'ACP), le troisième l'angle maximum de rotation, et, au final, les données normalisées, et le MST correspondant

Les exemples ci-dessus sont des tirages sinusoïdaux comme dans la section précédente, auxquels on a fait subir une rotation d'angle $-\pi/4$. Comme précédemment le premier graphe représente le MST sur les données centrées-réduites de manière "classique" (division par l'écart type). Sont ensuite présentés les pourcentages d'inertie expliquée pour chaque ACP afin d'avoir une indication sur le nombre d'axes (linéaires) nécessaires à la représentation des données , l'angle maximal de rotation (modulo $\pi/2$), le poids de chaque axe et, finalement, le MST à la dernière itération. Les indicateurs de convergence sont les parties 3 et 4 du graphique (l'angle maximal de rotation doit converger vers 0 et le poids de chaque axe vers 1).

On constate que, si l'effet saturation arrive plus vite avec la rotation, l'algorithme proposé reste efficace.

1.3.4 Les cartes de Kohonen et la normalisation

Il s'avère que les cartes de Kohonen sont elles aussi très sensibles à la normalisation (en fait pratiquement elles sont même plus sensibles que les graphes). Pour comprendre un peu pourquoi une telle sensibilité existe, on va rappeler rapidement l'algorithme :

- tirage aléatoire d'un point dans la base
- recherche du vecteur code le plus proche
- déplacement par homothétie du vecteur code le plus proche et de ses voisins vers le point tiré aléatoirement

Dans les cas à 0 voisins (algorithme de quantization ou des k-means aléatoires) les vecteurs codes d'une cellule se déplacent donc en moyenne vers le barycentre des points de la base inclus dans la cellule de Voronoï dudit vecteur code (cellule calculée sur la base de tous les vecteurs codes).

Si les échelles sur les axes ne sont pas "judicieuses", les cellules de Voronoï des vecteurs codes vont mal représenter la topologie des données et les vecteurs codes vont s'éloigner de leur place "optimale". Ce phénomène est illustré dans le graphique suivant (figure 1.12), 230 points ont été tirés suivant une sinusoïde, 200 points constituent les points de la base et 30 les vecteurs codes. Deux échelles ont été choisies. Dans le premier exemple, les flèches représentent de "mauvais" déplacements des vecteurs codes

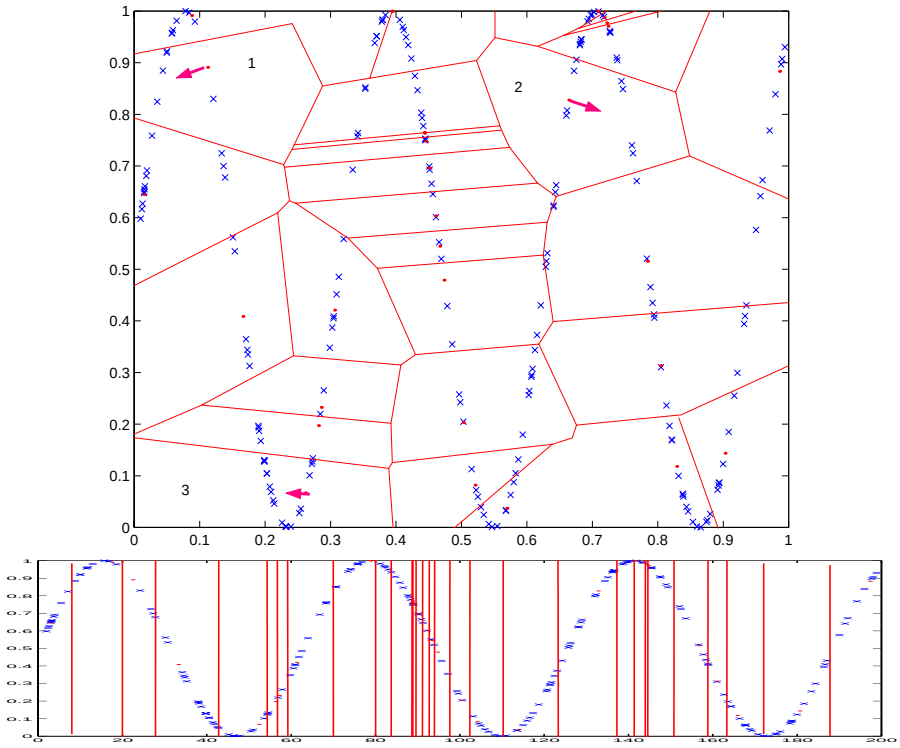


Fig. 1.12: Cellules de Voronoï pour deux exemples différant uniquement par les échelles et "mauvais" déplacements des vecteurs codes dans le premier exemple

(mauvais au sens où ces derniers s'éloignent du nuage de points). Dans le second exemple, la dilatation de l'axe horizontal est suffisante pour que les cellules de Voronoï soient des bandes verticales (ce qui représente la topologie des données : tout dépend de la variable x) et, si les vecteurs codes risquent de s'éloigner un peu des extrêma locaux (sous-estimation des maxima et sur-estimation des minima) ils restent dans l'ensemble corrects.

Le graphique suivant montre, pour les deux mêmes ensembles que précédemment, le résultat d'une ficelle de Kohonen de longueur 50.

Dans le premier cas, l'algorithme "confond" la sinusoïde avec un tirage uniforme et la ficelle ne représente pas du tout la "vraie" topologie des données alors que dans le second cas, on a convergence vers une ficelle ayant la même forme que les données.

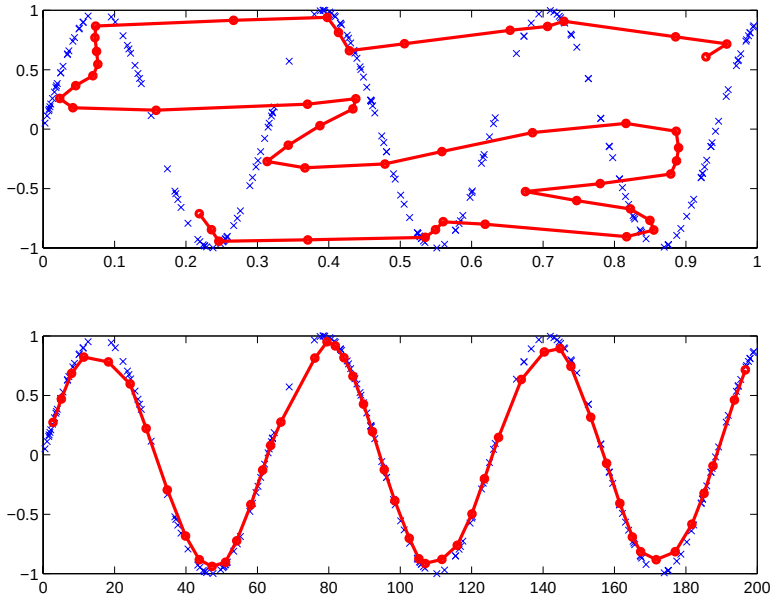


Fig. 1.13: Résultats de cartes de Kohonen pour deux mêmes ensembles à des échelles différentes

L'algorithme de normalisation proposé permet de donner des échelles sur les différents graphes dans une certaine mesure compatibles avec les cartes de Kohonen. Ci dessous les graphiques illustrent les résultat de ficelles de Kohonen sur des ensembles sinusoïdaux avant et après normalisation, apres 5000 itérations pour des ficelles de longueur variable (20,40 et 70).

Les exemples suivant illustrent les résultats de cartes de Kohonen sur des ensembles de dimension 2 (dimension intrinsèque) en dimension 3 (nombre d'axes linéaires nécessaires).

1.4 Conclusion

Les résultats des deux algorithmes de normalisation (simple et avec recherche des axes principaux) donnent des résultats qui vont au delà de nos espérances, mais la complexité des phénomènes en jeu dans l'algorithme rend les calculs théoriques ardus. Pour la normalisation simple, on a partir d'exemples qu'il n'y avait pas forcément d'existence d'une solution (c'est-à-dire d'un ensemble de poids qui rendent les tailles de liaisons égales à 1 en moyenne suivant toutes les directions). On a

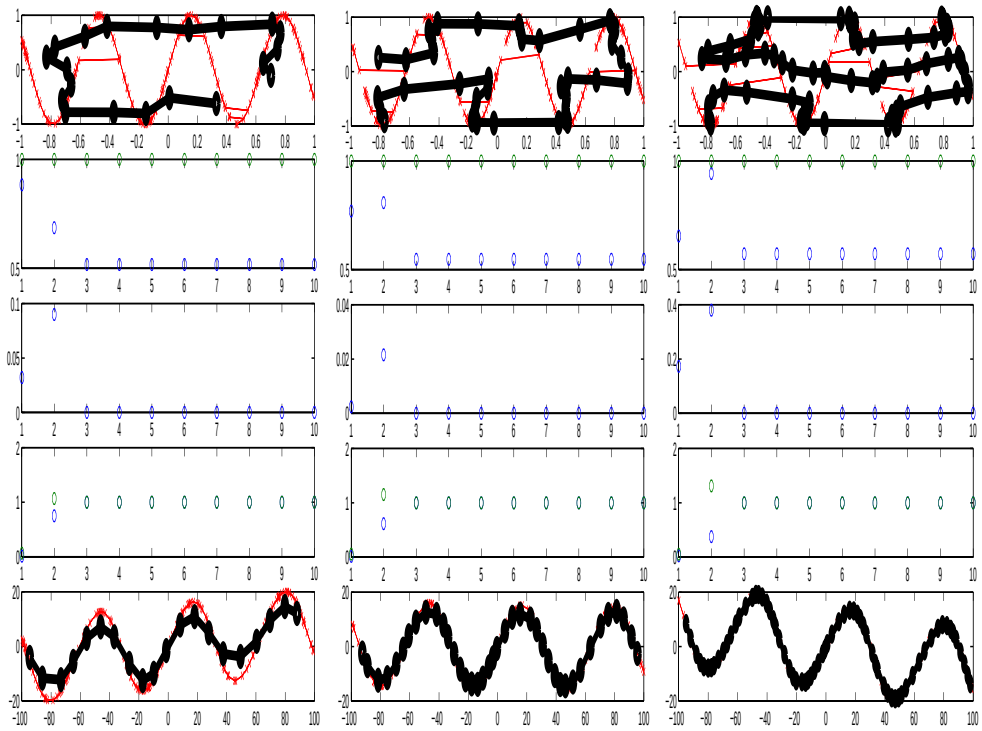


Fig. 1.14: Résultats avec des tirages de 200 points avec $X^1 = unifrnd(-1, 1)$ et $X^2 = sin(10X^1)$ on observe les performances d'une ficelle de Kohonen avec 20, 40 and 70 vecteurs codes et 5000 itérations (avant et après normalisation)

aussi trouvé des exemples tels que la solution ne soit pas unique. Dans le cas simple de tirages uniformes sur des rectangles, on a montré que c'était les "problèmes de bords" qui faisaient converger l'algorithme. On a montré ici de manière empirique que les résultats de la normalisation permettent un meilleur calcul de la distance curviligne et de meilleurs résultats pour des projections sur des cartes de Kohonen. On verra par la suite que la normalisation permet de mieux estimer les distances intrinsèques.

L'étude théorique a été abandonnée en raison de sa complexité mais si une piste simplificatrice se présente, elle sera reprise.

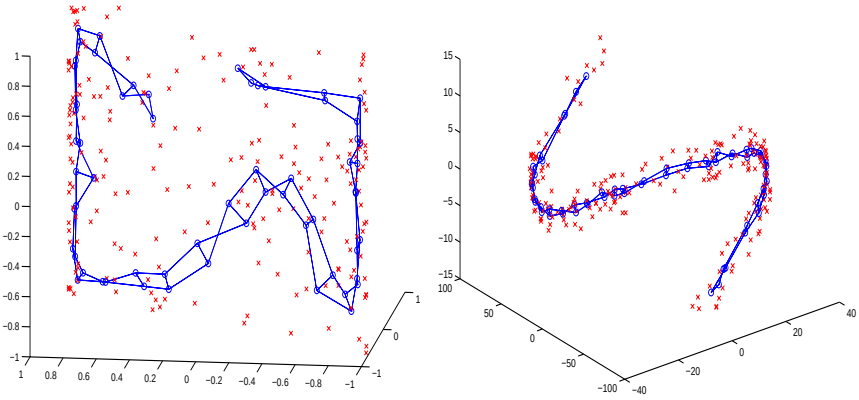


Fig. 1.15: Résultats avec des tirages de 200 points avec $X^1 = unifrnd(-1,1)$; $X^2 = sin(2X^1)$; $X^3 = unifrnd(-1,1)$ on observe les performances d'une carte de Kohonen (2,50) et 5000 itérations (avant et après normalisation)

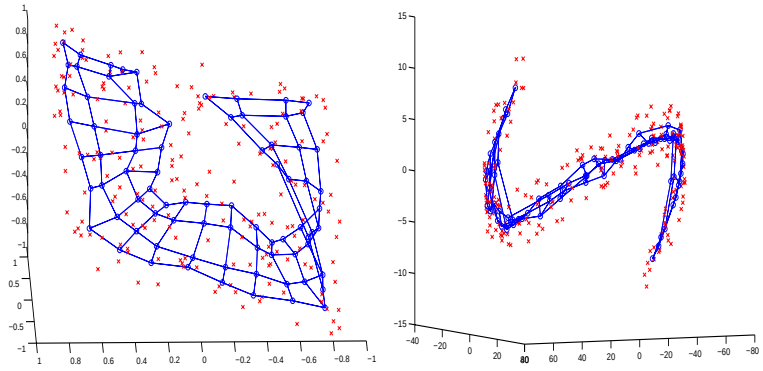


Fig. 1.16: même exemple que précédemment avec une carte $(4,20)$

2. CONSTRUCTION D'UN INDICATEUR CENTRAL

2.1 Principe et indicateur

Le barycentre G d'un ensemble de points peut être défini comme le point qui minimise l'inertie du nuage:

$$G = arg\min_X \{\sum_i \|X - X_i\|^2\}$$

De même, on peut définir le centre C d'un ensemble comme :

$$C = arg\min_X \{\max_i \|X - X_i\|^2\}$$

(il existe aussi le "mediancenter" M définit par : $C = arg\min_X\{\sum_i \|X - X_i\|\}$ et on pourra définir autant d'indicateur centraux qu'il existe de norme dans $\mathbb{R}^n$)

Le barycentre prend ainsi en compte les différences de répartition de masse dans l'ensemble, alors que le centre ne tient compte que de la "forme" de l'ensemble.

Des exemples de barycentres et de centres sont présentés dans la figure suivante.

Dans le second cas où les données sont réparties sur une forme parabolique, ni le barycentre ni le centre n'appartiennent à l'ensemble de points et, si ces deux indicateurs ont leur intérêt, on préférerait, dans une certaine mesure, avoir le sommet de la parabole comme indicateur central.

Pour cela il est aisé de définir un "barycentre connexe" et un "centre connexe" en adaptant les définitions précédentes en remplaçant la distance euclidienne par la distance curviligne.

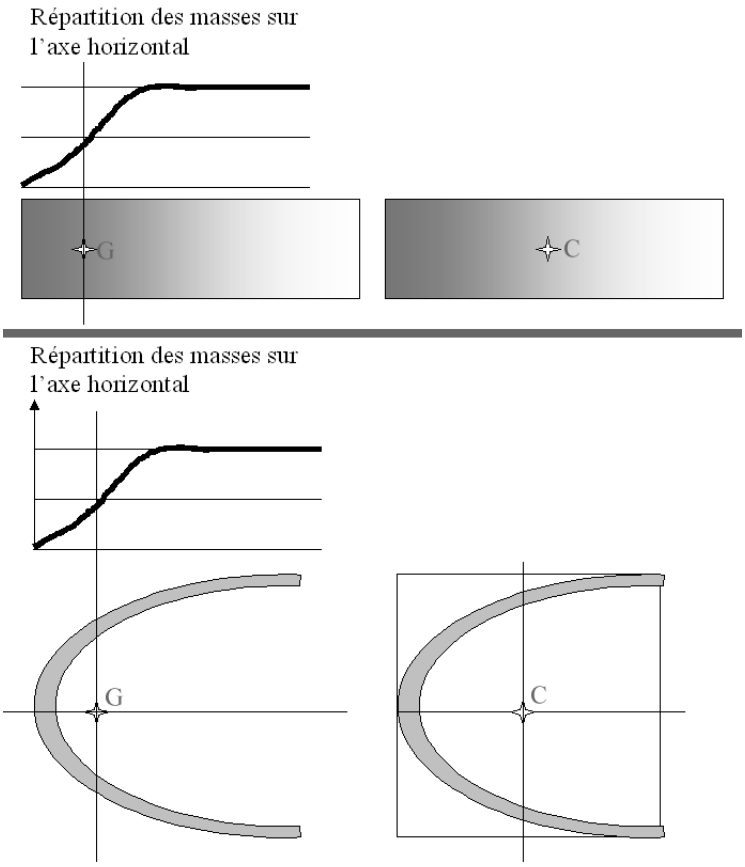


Fig. 2.1: Barycentre et centre pour deux ensembles

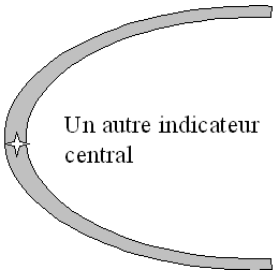


Fig. 2.2: Un indicateur central plus pertinent dans le cas de la parabole

Le "barycentre curviligne" est alors défini par :

$$G_c = arg\min_X \{\sum_i d_c(X - X_i)^2\}$$

et le "milieu curviligne" par :

$$C_c = arg\min_X \{\max_i d_c(X - X_i)^2\}$$

Où d_c est la distance curviligne.

D'un point de vue pratique on approchera les barycentre et centre curviligne par :

$$\overline{G_c} = arg\min_{X_j} \{\sum_i d_c(X_j - X_i)^2\}$$

et :

$$\overline{C_c} = arg\min_{X_j} \{\max_i d_c(X_j - X_i)^2\}$$

L'introduction de la distance curviligne, calculée sur l'ensemble de points, amène à un problème de non-unicité des deux indicateurs centraux G_c et C_c : en effet sur des ensembles présentant un "bouclage" (cercles, cylindres, sphères...) il n'existe pas un seul minimum aux fonctions $\sum_i d_c(M - X_i)^2$ et $\max_i d_c(M - X_i)^2$ mais un ensemble de minima.

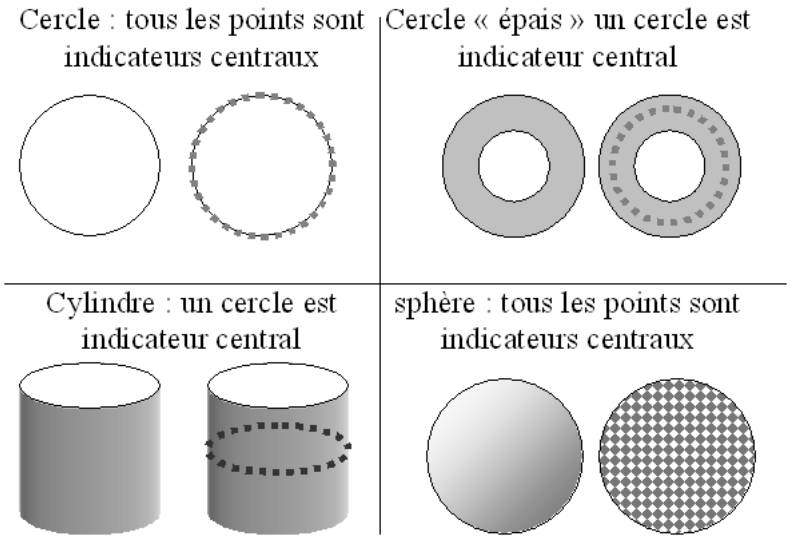


Fig. 2.3: Exemples d'ensembles pour lesquels les indicateurs centraux calculés avec la distance curviligne ne sont pas uniques

Pour éviter alors de choisir un point unique réalisant le minimum si ce dernier n'est pas significatif, on se propose d'afficher le diagramme

des $\sum_i d_c(M - X_i)^2$ ou des $\max_i d_c(M - X_i)^2$ triés, et de proposer à l'utilisateur de choisir un nombre de points parmi ceux réalisant les plus petites valeurs de ces fonctions. Sur ce nouvel ensemble Y de points on calculera alors les écarts-type suivant les différentes directions $(ect(\vec{Y}))$ que l'on comparera aux écarts-types de l'ensemble des observations suivant les différentes directions $(ect(\vec{X}))$ en observant les rapports entre les deux valeurs $(rap(i) = ect(Y_i)/ect(X_i))$. Sur les axes où ce rapport est faible (les variations de l'indicateur central sont faibles par rapport aux variations de l'ensemble des points) on choisira $G_{ci} = \overline{Y_i}$. En revanche sur les directions pour lesquelles les variations de Y sont comparables à celles de X on conservera toutes les valeurs de Y.

2.2 Résultats

D'un point de vue pratique, c'est le calcul du centre curviligne C_c qui a donné les meilleurs résultats. Dans le cas d'ensembles pour lesquels l'indicateur central est unique (pas de bouclage) les résultats sont équivalents pour les deux indicateurs. Par contre dans les cas de bouclages, la notion de centre a été bien plus performante. Ce sont donc les résultats obtenus avec le calcul du centre qui seront présentés ici.

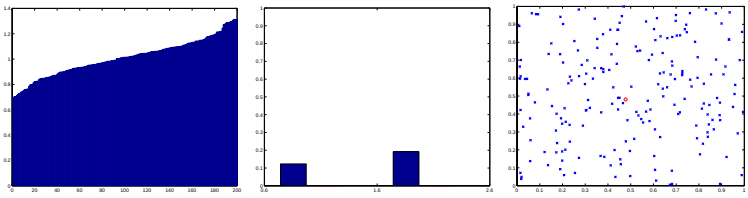


Fig. 2.4: Résultats pour un tirage uniforme de 200 points : sur une dizaine de points, l'écart type étant faible, on choisit de prendre le barycentre

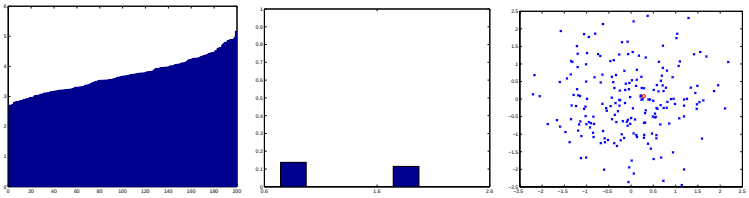


Fig. 2.5: Résultats pour un tirage gaussien de 200 points : sur une dizaine de points, l'écart type étant faible, on choisit de prendre le barycentre

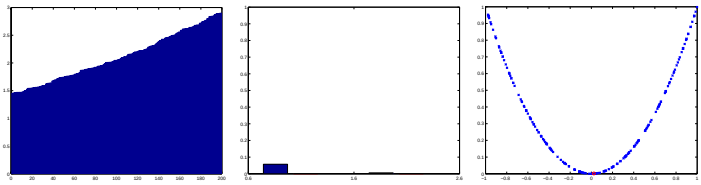


Fig. 2.6: Résultats pour un tirage sur une parabole de 200 points : sur une dizaine de points, l'écart type étant faible, on choisit de prendre le barycentre

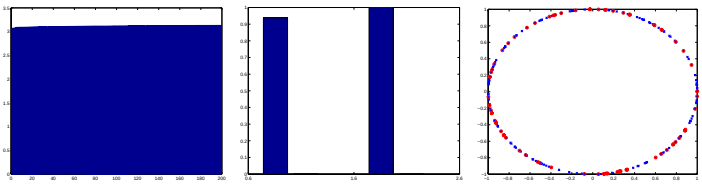


Fig. 2.7: Résultats pour un tirage uniforme sur un cercle de 200 points : sur 150 points, l'écart type étant presque égal à celui de la base, on conserve tous les points

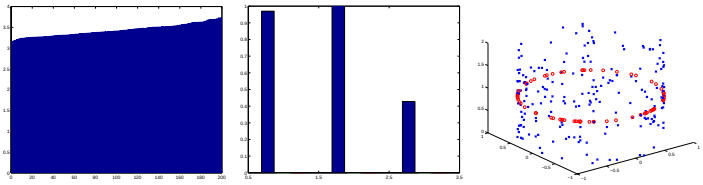


Fig. 2.8: Résultats pour un tirage uniforme sur un cylindre de 200 points : sur 100 points, l'écart type étant presque égal à celui de la base pour les axes 1 et 2, on conserve tous les points, en revanche on prend la moyenne pour le troisième axe

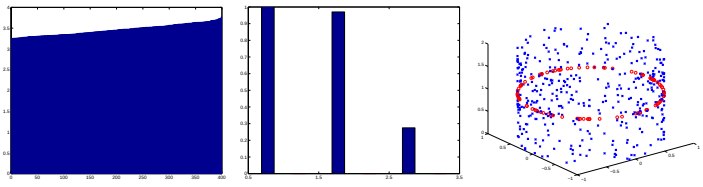


Fig. 2.9: Résultats pour un tirage uniforme sur un cylindre de 400 points : sur 100 points, l'écart type étant presque égal à celui de la base pour les axes 1 et 2, on conserve tous les points, en revanche on prend la moyenne pour le troisième axe

2.3 Perspectives

Dans un premier temps il faut trouver une méthode permettant de choisir le "nombre" d'indicateurs centraux à conserver en n'observant plus les dispersions sur les axes canoniques mais sur le système d'axe le plus pertinent en fonction de l'ensemble considéré, pour cela il nous faudra certainement, comme pour la normalisation, introduire une ACP dans l'algorithme.

Le choix de l'indicateur central est aussi à étudier plus en détail. On ne s'est concentré ici que sur le barycentre et le centre (la pratique nous a fait préféré le centre mais il nous faudrait aussi tester le "mediancenter" et, de manière générale tout indicateur possible).

Une première perspective, intéressante pour la suite de la thèse serait la mise en place d'un test d'existence de "boucles" dans les données. En résumé, lorsqu'il existe une "boucle" dans les données, l'indicateur central n'est pas unique. Bien sûr de telles considérations rapides ne sont pas rigoureuses et cette idée reste à développer. En effet dans le dernier chapitre de cette partie, on projettera les données sur des hyper-rectangles. Une telle projection ne peut donner de bons résultats (au sens du respect de la topologie) si le support des données n'est pas homéomorphe à un hyper-rectangle ce qui est évidemment le cas si il existe une boucle. Une deuxième perspective, serait la mise en place d'algorithmes de classification de type "classique" autour de ces centres (et non plus autour des barycentres).

3. ESTIMATION DE LA DIMENSION INTRINSÈQUE

La connaissance de la dimension intrinsèque d'un ensemble de points est une information fondamentale, d'une part pour pouvoir poursuivre l'analyse de l'ensemble, par des méthodes de projection non linéaires par exemple (dans ce cas la connaissance de la dimension permettra de déterminer le nombre d'axes "non linéaires"), d'autre part, comme cité en introduction, si l'objectif de l'étude est la modélisation d'un phénomène, la comparaison de la dimension des ensembles de variables explicatives et de l'ensemble produit (explicatives et expliquées) nous dira si la recherche d'un modèle est, ou non, vaine.

3.1 Les différentes méthodes théoriques de calcul de la dimension

3.1.1 Box Counting Dimension

Les méthodes de calcul de la dimension intrinsèque sont, initialement, issues de la théorie des fractales ([DIM3]). Soit X un ensemble de points, la première dimension mise au point, "capacity dimension" ou "box counting" notée D_{cap}, consiste en l'observation du nombre d'hypercubes $N(\varepsilon)$ de coté ε nécessaires au recouvrement de X (notion fondée sur l'autosimilarité attendue des ensembles) avec :

$$D_{cap} = \lim_{\varepsilon \to 0} -\frac{log(N(\varepsilon))}{log(\varepsilon)}$$

Lorsque la limite n'existe pas, on a recours à des passages aux limites supérieures et inférieures.

3.1.2 La dimension de corrélation

Etant donnée la difficulté pratique du calcul de $N(\varepsilon)$ (le temps explose rapidement) on a le plus souvent recours à la dimension de correlation

D_{cor}. Cette dimension repose sur le principe que, si on a un ensemble de dimension d, le nombre de paires de points distants d'au plus r est proportionnel à r^d. Elle a été développée simultanément dans ([DIM3] et [DIM4]).

Pratiquement :

Soit $S = \{x_1, ...\}$ un sous-ensemble dénombrable de X et $S_n = \{x_1, ..., x_n\}$ le sous-ensemble des premiers éléments de S. Soit alors :

$$C_n(r, S) = \frac{1}{n(n-1)} \sum_{i \neq j} 1_{\{d(x_i, x_j) < r\}}$$

$$C(r, S) = \lim_{n \to \infty} C_n(r, S)$$

$$D_{corr}(S) = \lim_{r \to 0} \frac{log(C(r, S))}{log(r)}$$

Dans les cas pratiques, la dimension de corrélation existera (i.e. ne dépend pas du sous-ensemble dénombrable S) dans les cas plus "exotiques", comme dans la "capacity" dimension on considérera les limites supérieures et inférieurs.

Il a été prouvé dans que si D_{cap} et D_{corr} existaient, on avait égalité de ces deux mesures. Dans la pratique, on préfère la seconde méthode car son coût algorithmique est bien moindre.

La dimension de corrélation a été généralisée en :

$$C_n^q(r, S) = [\frac{1}{n} \sum_{i \neq j} (\frac{1}{n-1} 1_{\{d(x_i, x_j) < r\}})^{q-1}]^{1/(q-1)}$$

$$C^q(r, S) = \lim_{n \to \infty} C_n^q(r, S)$$

$$D_{corr}^q(S) = \lim_{r \to 0} \frac{log(C^q(r, S))}{log(r)}$$

3.1.3 Les méthodes de Packing-number ou d'ensembles séparables

D'autres méthodes ont été récemment mises au point telle que les "packing number" ([DIM6]) qui reposent sur l'idée de Grassberger d'étudier le nombre de $q-$uplets de la base qui forment un ensemble séparable avec une distance de moins de r :

soit $G_q(N,r) = card\{(x_{i_1},...,x_{i_q}) \text{ tq } \forall n,m \in \{i_1,...,i_q\}^2 \; d(x_n,x_m) < r\}$

La dimension est alors calculée comme :

$$D_{q-point} = \frac{1}{q-1} \lim_{r\to 0} \lim_{N\to\infty} \frac{log(G_q(N,r))}{log(r)}$$

3.1.4 Sur les $k-$plus proches voisins

Il s'agit d'une méthode initiée par Guckenheimer et Buzyna dans [DIM5] puis améliorée, notamment par Baadi dans [DIM1] dont le principe repose sur l'évolution de la distance aux plus proches voisins.

Soit $d_k(x_i)$ la distance au k^{eme} plus proche voisin de x_i et : $\overline{d_k^\gamma} = \frac{1}{N}\sum_i d_k(x_i)^\gamma$ on a :

$$\overline{d_k^\gamma} \sim \frac{\Gamma(k+\gamma/D)}{\Gamma(k)} N^{-\gamma/D}$$

avec D la dimension.

Une telle équivalence entre des fonctions va permettre d'estimer la dimension.

3.2 L'estimation de dimension en pratique

La Box-Counting dimension qui, théoriquement est le "meilleur" indicateur, car le plus proche de la dimension topologique (même définition sur des recouvrements pour des ensembles infinis) pose, en pratique deux problèmes. Le premier, purement algorithmique, est celui du calcul de $N(\varepsilon)$ nombre minimal de cubes de coté ε nécessaires au recouvrement de l'ensemble des observations. On se limite au calcul du nombre de cubes de coté ε nécessaires au recouvrement en effectuant un pavage de l'espace et en comptant le nombre de cubes dans lesquels il y a au moins une observation. On obtient ainsi, non pas le nombre $N(\varepsilon)$, mais une quantité supérieure dépendant du point de départ du pavage. Un autre problème, plus délicat, inhérent à la méthode, est le passage à la limite lorsque l'ensemble de points dont on étudie la dimension est fini. En effet on avait défini la dimension de "Box Couning" par :

$$D_{cap} = \lim_{\varepsilon\to 0} -\frac{log(N(\varepsilon))}{log(\varepsilon)}$$

Lorsque le nombre de points est fini, il est bien évident que :

$$\lim_{\varepsilon \to 0} -\frac{log(N(\varepsilon))}{log(\varepsilon)} = 0$$

Remarque : ceci est tout-à-fait cohérent étant donné que la dimension topologique d'un ensemble de point est 0.

Dans la pratique on observe le nuage de points constitué des couples $(N(\varepsilon), \varepsilon)$ sur une échelle logarithmique et on calcule la pente sur un domaine où la courbe est droite. Vient alors le problème de la définition d'un domaine où la courbe est une droite...

L'estimation de la dimension de correlation (et de sa généralisation) dans le cadre de données finies pose le même type de problème. On avait :

$$C_n(r, S) = \frac{1}{n(n-1)} \sum_{i \neq j} 1_{\{d(x_i, x_j) < r\}}$$

$$C(r, S) = \lim_{n \to \infty} C_n(r, S)$$

$$D_{corr}(S) = \lim_{r \to 0} \frac{log(C(r, S))}{log(r)}$$

Etant donné qu'on dispose d'un nombre fini d'observations on devra, en pratique se contenter de :

$$C_N(r) = \frac{1}{N(N-1)} \sum_{i \neq j} 1_{\{d(x_i, x_j) < r\}}$$

dont on ne pourra prendre la limite, ni en N, ni en r.

Ici aussi on devra se contenter de la lecture du nuage de points de $log(C_N(r))$ en fonction de $log(r)$ et d'en prendre la pente sur une plage suffisamment "linéaire". Un seuil qui peut sembler "raisonnable" pour mesurer la pente de $log(C_N(r))$ en fonction de $log(r)$ est le seuil de connexité δ. L'idée étant que d'une part ce seuil est supérieur à la plus grande des plus petites distances entre deux points. En effet toutes les boules centrées sur un point des observations x_i et de rayon δ comprendront au moins un $x_{j \neq i}$. Ce ne sera donc pas un seuil "trop petit" pour une limite vers 0. D'autre part l'ensemble étant $\delta-$connexe on ne risque pas d'agréger des mesures de dimensions de sous parties connexes.

Enfin rappelons que la dimension de corrélation correspond à une hypothèse de tirage uniforme et fonctionne convenablement si la densité

sous jacente n'est pas trop éloignée de la loi uniforme.

La méthode des $k-$plus proches voisins a l'avantage quant à elle de ne recourir à aucune limite (excepté en N).

Enfin après avoir listé les inconvénients des diverses méthodes on peut noter que, dans le cas de l'analyse des données, on s'attend à trouver des dimensions entières. Au lieu de calculer une dimension on recherchera donc la dimension entière collant le mieux aux données.

Les deux méthodes que nous avons choisies sont : la dimension de corrélation, pour tester et observer la pertinence du choix du seuil de connexité pour son estimation, et la méthode des $k-$plus proches voisins car elle s'adapte bien à la recherche de la "meilleure" dimension entière.

3.3 La dimension de correlation autour du seuil de connexité

On va observer le graphique de :

$$log(C_N(r)) = log(\frac{1}{N(N-1)} \sum_{i \neq j} 1_{\{d(x_i,x_j)<r\}})$$

en fonction de $log(r)$.

Pour estimer la dimension, on placera sur le graphique les droites de pentes $1, 2, ...d$ passant par le point $(log(\delta), log(C_N(\delta)))$ où δ est le seuil de connexité de l'ensemble des observations (X). La lecture de la dimension se fera par l'observation du résultat graphique. On choisira parmi l'ensemble des pentes celle qui semble "coller" le mieux à la courbe (dans un premier temps on ne mettra pas en place de critère des moindres carrés car on ne sait pas, a priori, sur combien de points le calculer).

Plusieurs problèmes apparaissent qui rendent l'estimation de dimension relativement peu fiable pour des grandes dimensions : d'une part le nombre de points nécessaire à une bonne "représentation" de la dimension d croît en N^d. Par exemple en observant sur les graphiques suivants qu'une dimension 3 est à peu près retrouvée pour des tirages de 100 points (mais pas une dimension 4), on en déduit que le nombre de points nécessaires à l'estimation d'une dimension D est supérieur à 4^D

$(100^{1/3} \sim 4,5)$ ce qui donne un nombre de points nécessaires supérieur à 256 pour $D = 4$, 1024 pour $D = 5$, 4096 pour $D = 6$, 16384 pour $D = 7$.... D'autre part, lorsque d croît les droites de pentes d et $d+1$ se rapprochent les unes des autres.

Enfin la méthode ayant été construite pour des tirages de type "uniforme" et comme on veut seulement tester l'idée intuitive de l'estimation de densité au seuil de connexité, on examinera les résultats pour des tirages de type uniformes.

3.3.1 Données uniformes sur $[0,1]^d$

Pour 100, 500 et 2000 points on s'attend à retrouver des dimensions 3, 4 et 5, mais pas au delà.

Les données ont été simulées sur $[0,1]^d$ et les dimensions testées de 1 à d.

On observe d'une part que la capacité à retrouver la dimension correspond bien aux attentes en nombre de points et, d'autre part que l'idée de se placer au seuil de connexité pour l'estimation de la pente semble intéressante.

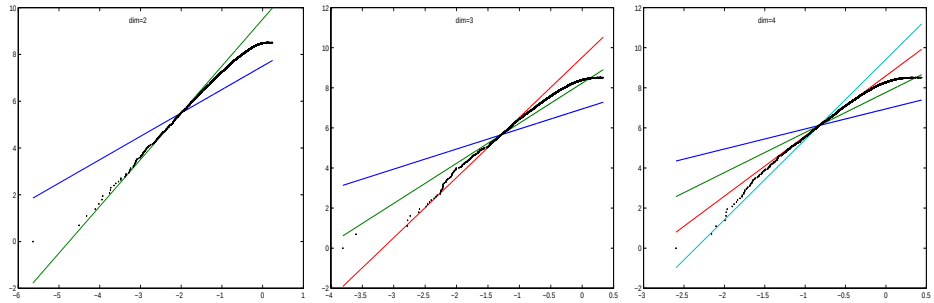


Fig. 3.1: Estimation de la dimension au seuil de connexité pour des tirages uniformes de 100 points

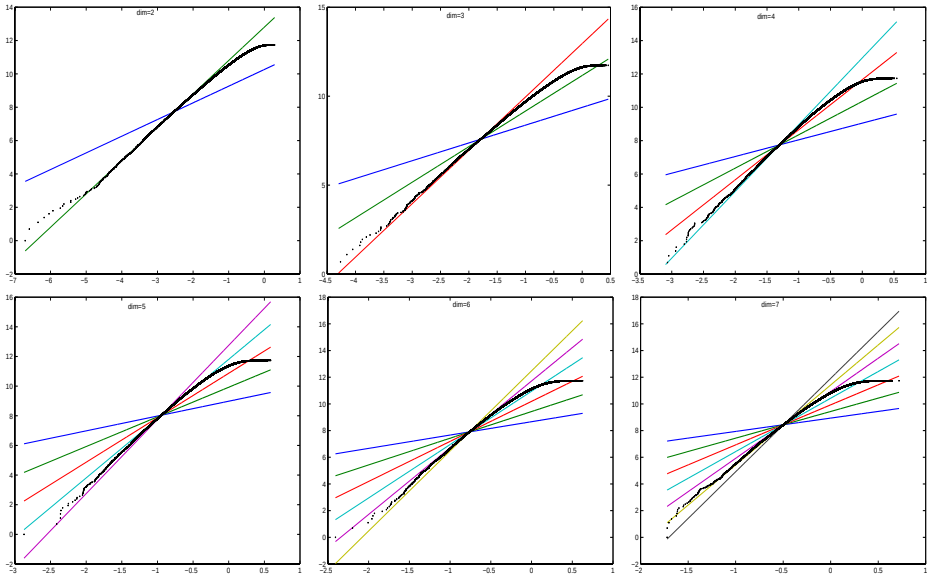


Fig. 3.2: Estimation de la dimension au seuil de connexité pour des tirages uniformes de 500 points

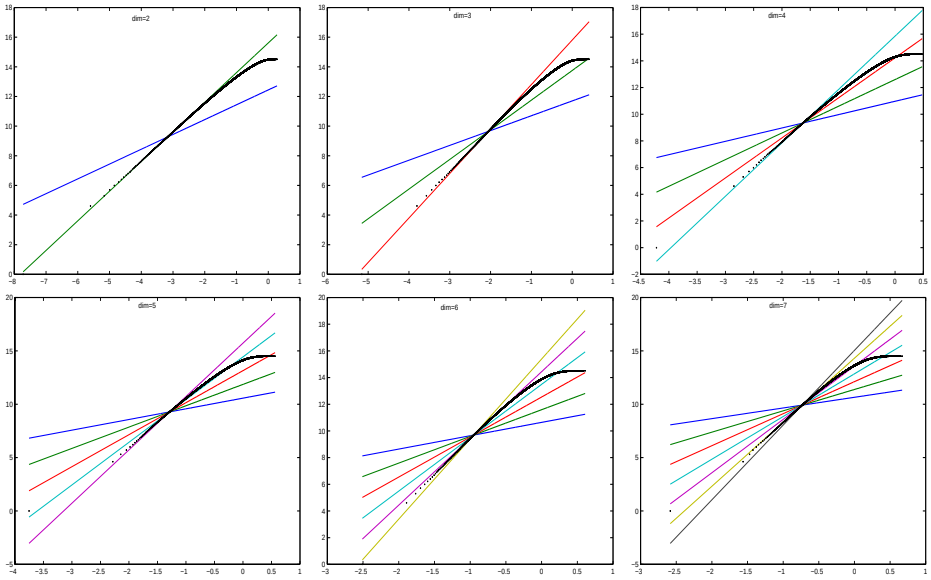


Fig. 3.3: Estimation de la dimension au seuil de connexité pour des tirages uniformes de 2000 points

3.3.2 Normalisation et estimation de la dimension

Soulignons une fois de plus l'importance de la normalisation des données pour l'estimation de densité. Encore une fois on a choisi un exemple "sinusoïdal", soit, théoriquement un ensemble de dimension une, non linéaire, plongé en dimension deux. Les graphiques du MST et de l'évolution de $(log(C_N(r))$ en fonction de $log(r)$ sont tracés pour deux ensembles, avant et après normalisation.

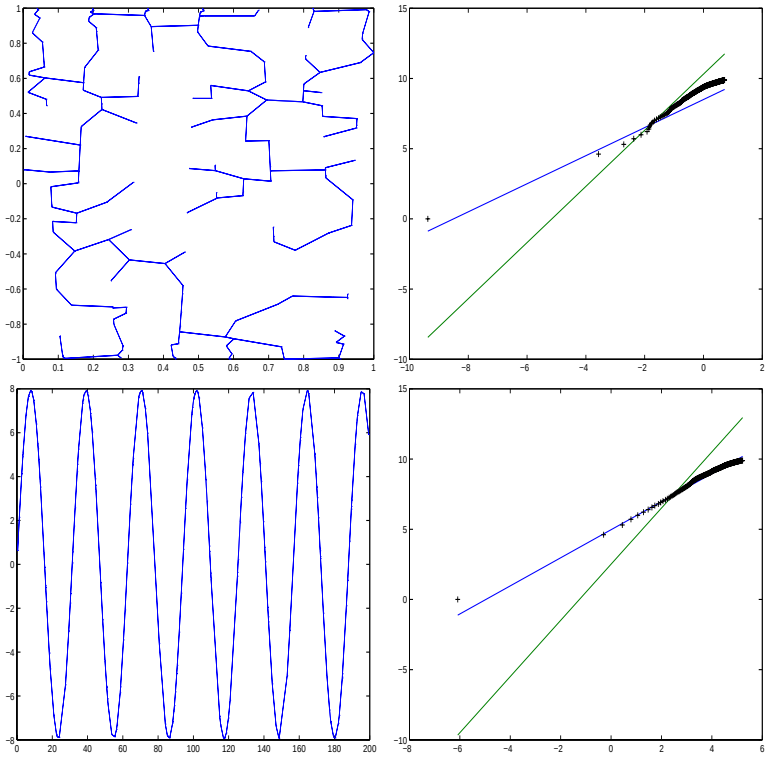


Fig. 3.4: Estimation de dimension sur une sinusoïde avant et après normalisation

Si le graphique après normalisation indique clairement une dimension 1 les choses sont moins claires avant normalisation où l'on semble observer une rupture de pente autour du seuil de connexité avec un passage de la dimension 1 à la dimension 2, la normalisation facilite ainsi la lecture et l'estimation de dimension.

3.4 La dimension des k−plus proches voisins

3.4.1 Méthode

Rappelons qu'on part de l'équivalence suivante :

Soit $d_k(x_i)$ la distance au k^{eme} plus proche voisin de x_i et : $\overline{d_k^\gamma} = \frac{1}{N}\sum_i d_k(x_i)^\gamma$ on a :

$$\overline{d_k^\gamma} \sim \frac{\Gamma(k+\gamma/D)}{\Gamma(k)} N^{-\gamma/D}$$

On remarque que cette équivalence n'est vraie que pour des tirages uniformes dans $[0,1]^D$, dans le cas de tirages uniforme dans d'autres espaces on aura :

$$\overline{d_k^\gamma} \sim \frac{\Gamma(k+\gamma/D)}{\Gamma(k)} C^{-\gamma/D}$$

où C est une constante.

On partira alors de cette dernière équation et comme on suppose la dimension entière, il suffit de tester toutes les valeurs entières de D inférieures ou égales au nombre de variables.

Comme l'équivalence est vraie pour n'importe quelle valeur de k et n'importe quelle valeur de γ, on va choisir de calculer $\overline{d_k\gamma}$ pour $k \in \{1,2,..,10\}$ et $\gamma \in \{0.1, 0.2, .., 10\}$.

Pour chaque valeur de D on recherchera C_D la constante maximisant la corrélation entre $\overline{d_k^\gamma}$ et $\frac{\Gamma(k+\gamma/D)}{\Gamma(k)} C^{-\gamma/D}$ et on notera $cor(D)$ le coefficient de corrélation entre $\overline{d_k^\gamma}$ et $\frac{\Gamma(k+\gamma/D)}{\Gamma(k)} C_D^{-\gamma/D}$. On estimera la dimension par $\overline{D} = argmax(cor(d))$

3.4.2 Résultats

On présente les histogrames de $cor(d)$ pour des tirages de 100 à 1000 points (en ligne) et des dimensions de 1 à 4 (en colonne) plongées en dimension 10. On constate qu'il faut 300 points pour commencer à retrouver la dimension 4 et (on n'a pas présenté les graphiques) et que la dimension 5 n'est pas retrouvée avant 1000 points. On a donc des ordres de grandeur des nombres de points nécessaires à l'estimation des dimensions équivalents à ceux de la méthode précédente. Le principal avantage de la méthode des $k-$plus proches voisins réside dans le fait qu'il n'y a pas de seuil à paramétrer pour l'estimation. Dans le graphiques suivant

(figure 3.5) on présente les résultats d'estimation de la dimension intrinsèque par la méthode des $k-$plus proches voisins, données simulées 1 première colonne avec des tirages de 100, 200,..,1000 points. La seconde colonne montre les résultats pour des données simulées en dimension 2,..., la quatrième correspond à des données simulées en dimension 4. Chaque graphique représente Pour chaque exemple le diagramme présente les coefficient de corrélation entre $\overline{d_k \gamma}$ et $\frac{\Gamma(k+\gamma/D)}{\Gamma(k)} C_D^{-\gamma/D}$ pour des valeurs de D variant de 1 à 10, la dimension estimée sera celle qui réalise le maximum de corrélation. De manière théorique on devrait donc avoir, pour la colonne k des diagrammes atteignant leur maximum pour une dimension testée égale à k.

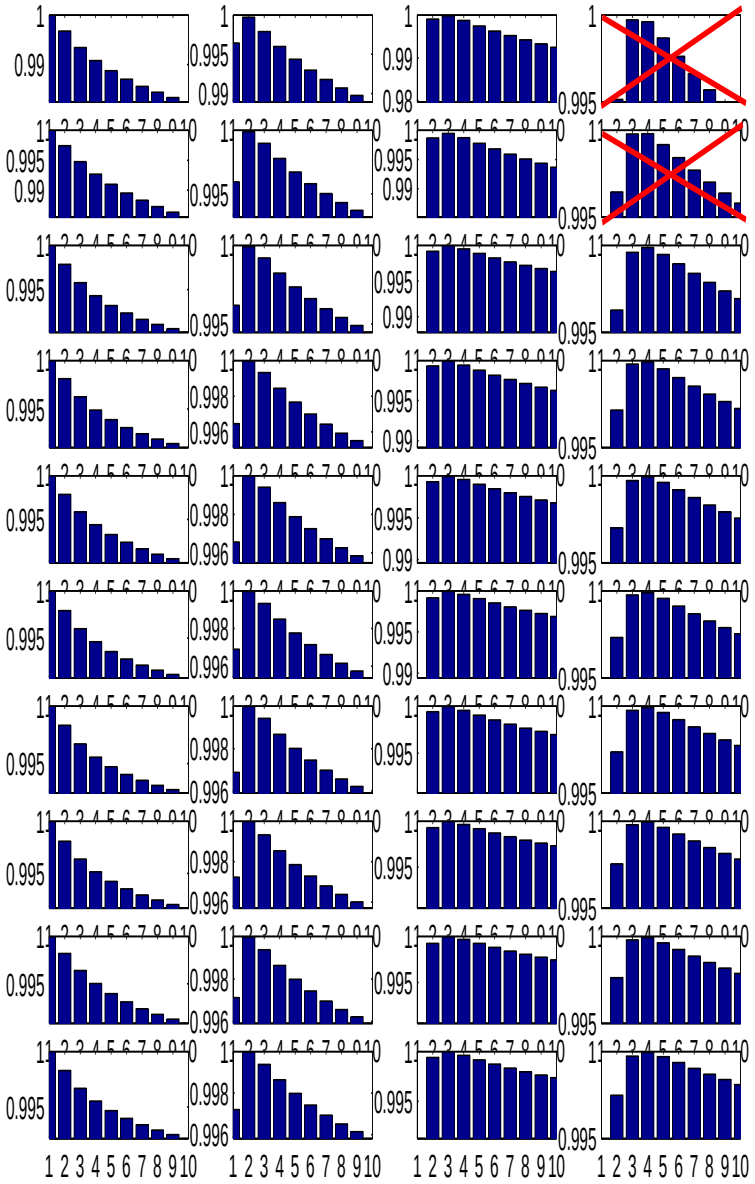


Fig. 3.5: Dimension intrinsèque par la méthode des $k-$plus proches voisins

3.4.3 Normalisation et dimension

Encore une fois la normalisation des données est nécessaire à une bonne estimation de la densité.

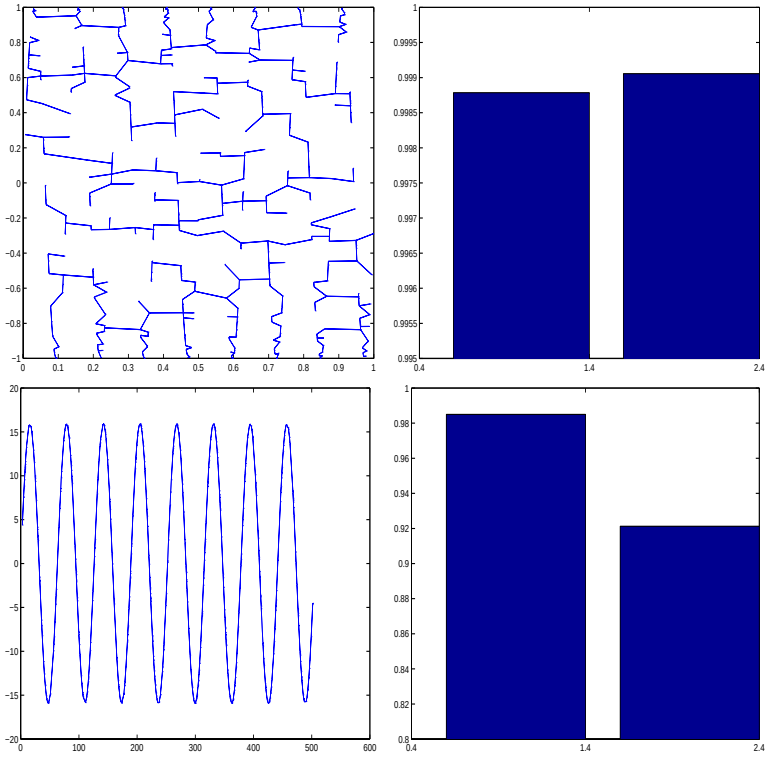


Fig. 3.6: Test des dimension 1 contre 2 par la méthode des $k-$plus proches voisins. Dans le premier cas (données normalisées de manière "classique") le meilleur coefficient de corrélation est réalisé pour une dimension estimée à 2, dans le second cas on choisit une dimension 1.

4. LES MÉTHODES DE PROJECTION

4.1 Les cartes de Kohonen

Bien que d'autres méthodes de projection non linéaires des données existent (telles qu'isomap, curvilinear distance analysis...), nous avons choisi ici de nous concentrer sur les cartes de Kohonen à voisinage "carré" car cette méthode, comme on le verra par la suite, s'adapte parfaitement à notre double objectif de consolidation de l'estimation de dimension et de finalisation (pratique) du test de connexité.

4.2 Le paramètrage des cartes de Kohonen

4.2.1 Introduction

Dans cette partie, on suppose qu'on a un ensemble X de N observations dans $\mathbb{R}^D$ $X_i \in R^D$ avec, pour caractériser le i^{eme} individu, les D variables $X_i = (X_i^1, ..., X_i^D)$.

On supposera ici, que les variables sont tirées aléatoirement sur une variété de dimension $d \leq D$ (ceci peut être dû à des liaisons internes entre les variables) et on supposera, surtout, que la variété est homéomorphe à un hyper-rectangle (ceci exclus, entre autre toute surface type cercle, tore...)

En lançant une carte de Kohonen sur les données on peut espérer observer l'organisation des données sur l'hyper rectangle ; tout le problème consistant à trouver une valeur correcte pour d et un nombre de cellule adéquat dans les d directions de la carte.

Pour cela on va construire un indicateur de préservation de la topologie (entre données initiales et leur projection sur les vecteurs codes).

4.2.2 Les mesures de préservation de la topologie

état de l'art

De nombreux travaux ont donné lieu à des indicateurs de préservation de la topologie qui ont été répertoriés par Goodhill et Sejnowski dans [TOP6]. Pour les lister de manière ordonnés les auteurs commencent par donner les indicateurs construits selon une $C-$mesure puis les "autres"

les mesures de préservation de la topologie reposant sur une $C-$mesure ont tous le même principe :

Soit F une matrice de similarité ou de dissimilarité sur l'ensemble des points des observations (X).

Soit $cl(i)$ la classe dans laquelle le point X_i est projeté à l'issue de la carte de Kohonen

Soit G une matrice de similarité ou de dissimilarité sur les centres des cases de la carte (qui va donc dépendre fortement de la structure topologique donnée en entrée).

Soit enfin

$$C = \sum_{i=1}^{N} \sum_{j<i} F(i,j)G(cl(i), cl(j))$$

La maximisation de C (dans le cas où F et G sont deux similarités ou deux dissimilarités) ou sa minimisation dans le cas où on a des matrices de natures différentes est une approximation de la maximisation (respectivement la minimisation) du coefficient de corrélation entre les similarités après convergence de l'algorithme de Kohonen.

A la place de C, on peut aussi observer le coefficient de correlation, et, bien sûr observer le nuage de points correspondant afin d'avoir une idée de la pertinence de l'indicateur.

Les exemples de couples F et G que l'on trouve dans la littérature sont :

- A- minimal writing [TOP6]:

- $F(i,j) = 1$ si X_i et X_j sont voisins, 0 sinon
- $G(cl(i), cl(j)) = ||cl(i), cl(j)||$

- B- minimal path length [TOP6]:
  - $F(i,j) = ||X_i - X_j||$
  - $G(cl(i), cl(j)) = 1$ si X_i et X_j sont voisins, 0 sinon

- C- Jones et al [TOP6]:
  - $F(i,j) = 1$ si X_i et X_j sont voisins, 0 sinon
  - $G(cl(i), cl(j)) = 1$ si X_i et X_j sont voisins, 0 sinon

Il existe d'autres mesures non fondées sur une $C-$ mesure :

D : Demartines et Hèraut [TOP3] construisent un indicateur de respect de la topologie en "dépliant" les données à partir des résultats de la carte (ils considèrent les lignes médianes de la carte comme des axes non linéaires et déplient les points de la base suivant ces axes). Une fois les observations "dépliées", ils observent les correlations entre les distances euclidiennes des points (dépliés) et les distances euclidiennes des vecteurs codes associés.

Une telle mesure de préservation de la topologie peut être elle-aussi considérée comme une $C-$ mesure avec :

- $F(i,j) = ||deplie(X_i), deplie(X_j)||$
- $G(cl(i), cl(j)) = ||cl(i), cl(j)||$

E : Une autre mesure, radicalement différente a été développée par Villmann et al. [TOP4] Pour quantifier la préservation de la topologie ils construisent une "topographic function " pour mesurer la similitude entre les voisinages dans l'espace des observations (construit à l'aide des cellules de Voronoï) et dans l'espace de projection.

D'autres indicateurs existent tel que la "minimal distortion" ou la mesure du STRESS [TOP2].

Une autre approche

Pourquoi une autre approche ? Si les méthodes A, B, C semblent relativement intuitives, elles ne reposent pas sur une définition mathématique de la préservation de la topologie. Par exemple les similarités ne reposant que sur une notion de voisinage codé en 0 ou 1 ne prennent en compte qu'une conservation très "locale" de la topologie.

La méthode E (de Villmann et al) repose sur une considération mathématique mais comporte quelques points faibles pratiques tels que le fait que les voisinages sont construits à partir des cellules de Voronoï, ce qui rend le calcul inenvisageable en dimension supérieure (ou égale) à 3 (notamment à cause du temps de calcul).

La méthode D est relativement proche de celle que nous allons proposer, la principale critique que l'on peut faire consiste à dire que le choix des axes médians de la carte pour déplier les observations sous-entend que la carte a convergé vers un résultat "correct" avant de juger de la qualité du résultat. Un autre problème réside dans le fait que, si on est bien capable à l'aide de cette méthode, de déterminer la dimension intrinsèque des données, on ne sera guère capable de différencier deux cartes de dimension 2 avec un nombre d'unités différent.

La méthode que nous allons proposer repose sur des considérations mathématiques de préservation de la topologie, mais reste proche en principe des $C-$mesures. Elle donne des résultats relativement similaires à ceux de Desmartines (mais aisément adaptables à des dimensions supérieures à deux et à des structures de carte différentes).

Caractérisation de la préservation de la topologie Soient (E_1, d_1) et (E_2, d_2) deux espaces métriques. Ils sont dis C_0 homéomorphes si il existe $g : E_1 \to E_2$ une bijection continue telle que $\forall (x, y) \in E_1^2, d_1(x, y) = d_2(g(x), g(y))$

Si (E_1) est une variété de dimension d que l'on veut rendre homéomorphe à un hyper-rectangle la distance naturelle à choisir sur E_1 est la distance géodesique.

Application aux cartes de Kohonen Dans notre problématique, l'espace des observations est le second espace de la carte de Kohonen, celui où les unités prennent leur valeur (vecteur code) .

Etant donné que les vecteurs codes donnent une quantification vectorielle de l'espace des observations, on se concentre essentiellement sur ces points et on observe la liaison entre la distance curviligne entre vecteurs codes sur l'espace des observations et la distance euclidienne entre les cases de la carte correspondante.

En résumé, on étudie la corrélation entre :

$d_1((i_1, ..., i_d), (j_1, ..., j_d)) = \sqrt{\sum (i_k - j_k)^2}$ et :

$d_2(w_{i_1,...,i_d}, w_{j_1,...,j_d})$ la distance curviligne entre les deux vecteurs code correspondants.

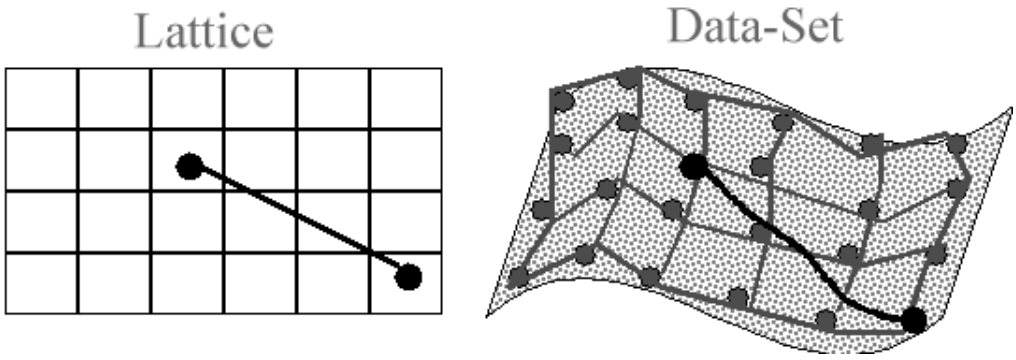


Fig. 4.1: Distance sur la carte et la distance correspondante entre les données.

Pour le calcul proprement dit de la distance curviligne, on peut proposer plusieurs méthodes qui dépendent essentiellement du nombre d'observations et du temps de calcul dont on dispose. Le calcul de la distance curviligne se fait en choisissant un graphe donnant les voisinages et les distances entre points si ils sont voisins, et en utilisant l'algorithme de Dijkstra.

L'algorithme de Didjkstra est relativement coûteux en temps et on propose plusieurs méthodes pour calculer le graphe.

- **Méthode exhaustive :** on se donne un nombre k de voisins et on utilise le graphe des $k-$plus proches voisins sur l'ensemble des observations et des vecteurs codes

- **Méthode simple :** on se donne un nombre k de voisins et on utilise le graphe des $k-$plus proches voisins sur l'ensemble des vecteurs codes

- **Graphe type** GNN (gaz de neurones) on calcule un graphe sur les vecteurs codes en s'inspirant de la méthode d'apprentissage des GNN : pour tous les points d'observations on ajoute la connexion entre les deux vecteurs codes les plus proches de l'observation. On consolide ensuite les liaisons en liant chaque vecteur code à ceux qui lui sont plus proches que les connexions données par la première étape

Si l'approche exhaustive semble a priori la meilleure, elle nécessite, en pratique un temps de calcul bien trop long. L'absence de paramétrage initial de la méthode inspirée des gaz de neurones, et le fait que le graphe soit construit à partir des observations nous fait préférer cette dernière approche.

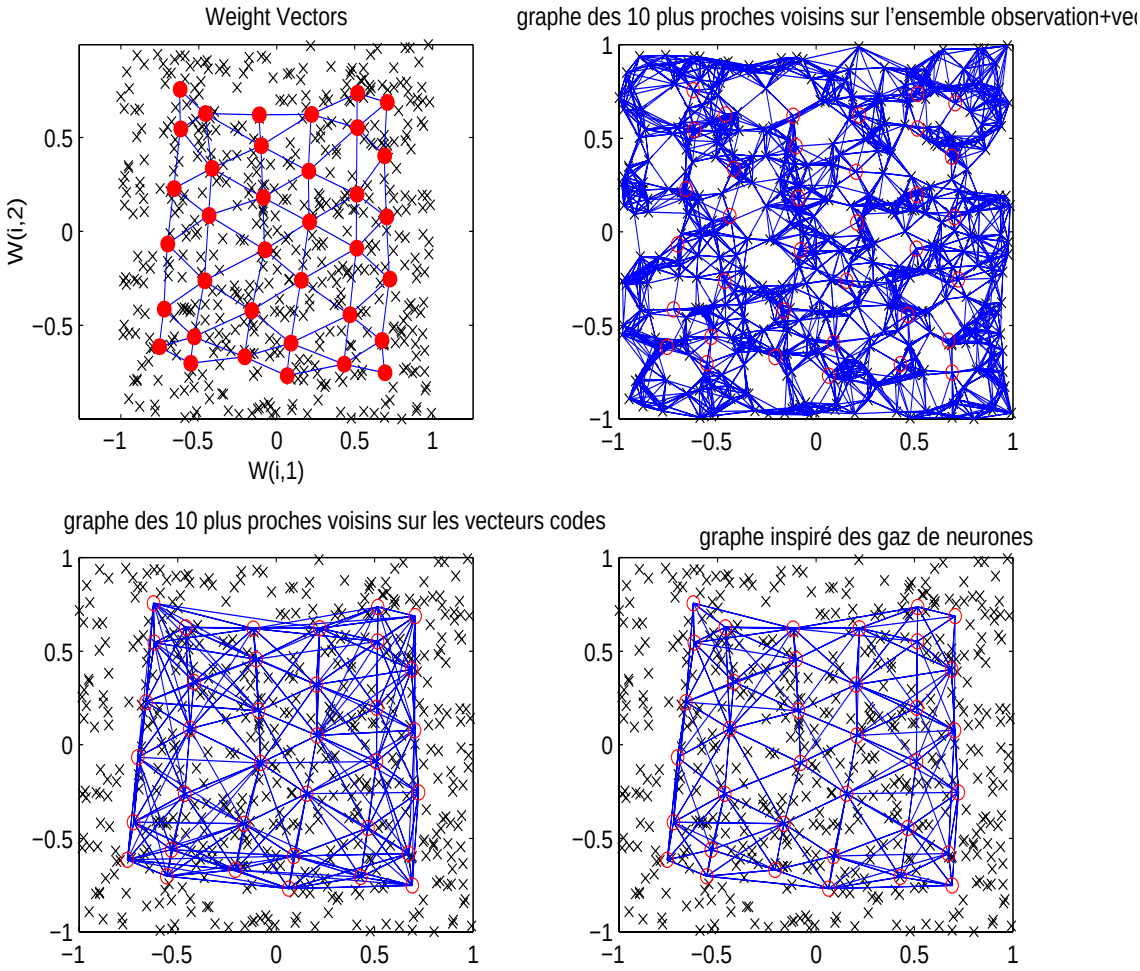


Fig. 4.2: Illustration des différentes méthodes de calcul des graphes.

4.2.3 Quelques résultats

"Meilleure" carte de Kohonen pour différents exemples

On a testé l'indicateur de préservation de la topologie sur différents exemples :

- A : 300 points uniformément tirés sur $[0,1]^2$
- B : 300 points uniformément tirés sur une selle de cheval
- C : 300 points tirés sur une ligne en dimension 3

- D : 300 points tirés sur une sinusoïde de fréquence 50 (normalisés par la méthode précédemment explicitée)
- A' : 300 points uniformément tirés sur $[0,1]^2$ et bruités sur une troisième dimension
- B' : 300 points uniformément tirés sur une selle de cheval et bruités sur une quatrième dimension
- C' : 300 points tirés sur une ligne en dimension 3 et bruités sur une quatrième dimension
- D' : 300 points tirés sur une sinusoïde de fréquence 50 et bruités sur l'axe y (normés)
- E : enfin un exemple ne correspondant pas aux hypothèses de tirage homéomorphe à un hyper rectangle : un tirage uniforme de 300 points sur un cercle

On teste 10 cartes de Kohonen ayant approximativement 100 vecteurs codes ($(1,100)$, $(2,50)$, $(3,32)$,...,$(10,10)$), numérotées de 1 à 10 suivant le nombre de lignes. Trois graphiques présentent les résultats pour chaque exemples (figures 4.3, 4.4 et 4.5):

- Les 10 nuages des couples donnés par les distances curvilignes estimées à l'aide du graphe inspiré des GNN (axe des abscisses) et les distances euclidiennes entre les cases correspondantes.
- Le coefficient de corrélation entre les deux distances.
- La carte pour le maximum de corrélation.

On remarque alors que les résultats sont ceux attendus, on retrouve des cartes "carrées" meilleures dans les cas des tirages A,B,A' et B' et des ficelles dans les cas C,D, C' et D'.

Dans le cas de l'exemple E, qui ne satisfait pas aux hypothèses de travail, l'étude du nuage de points est plus instructive que celle du coefficient de corrélation : Même si ce dernier indique une ficelle (et donc une dimension intrinsèque de 1) il est croissant puis décroissant et, dans certain cas indique plutôt une structure "carrée". En revanche la

lecture du nuage de points montre une relation étroite entre les deux distances dans le cas d'une ficelle (même si la relation n'est pas linéaire).

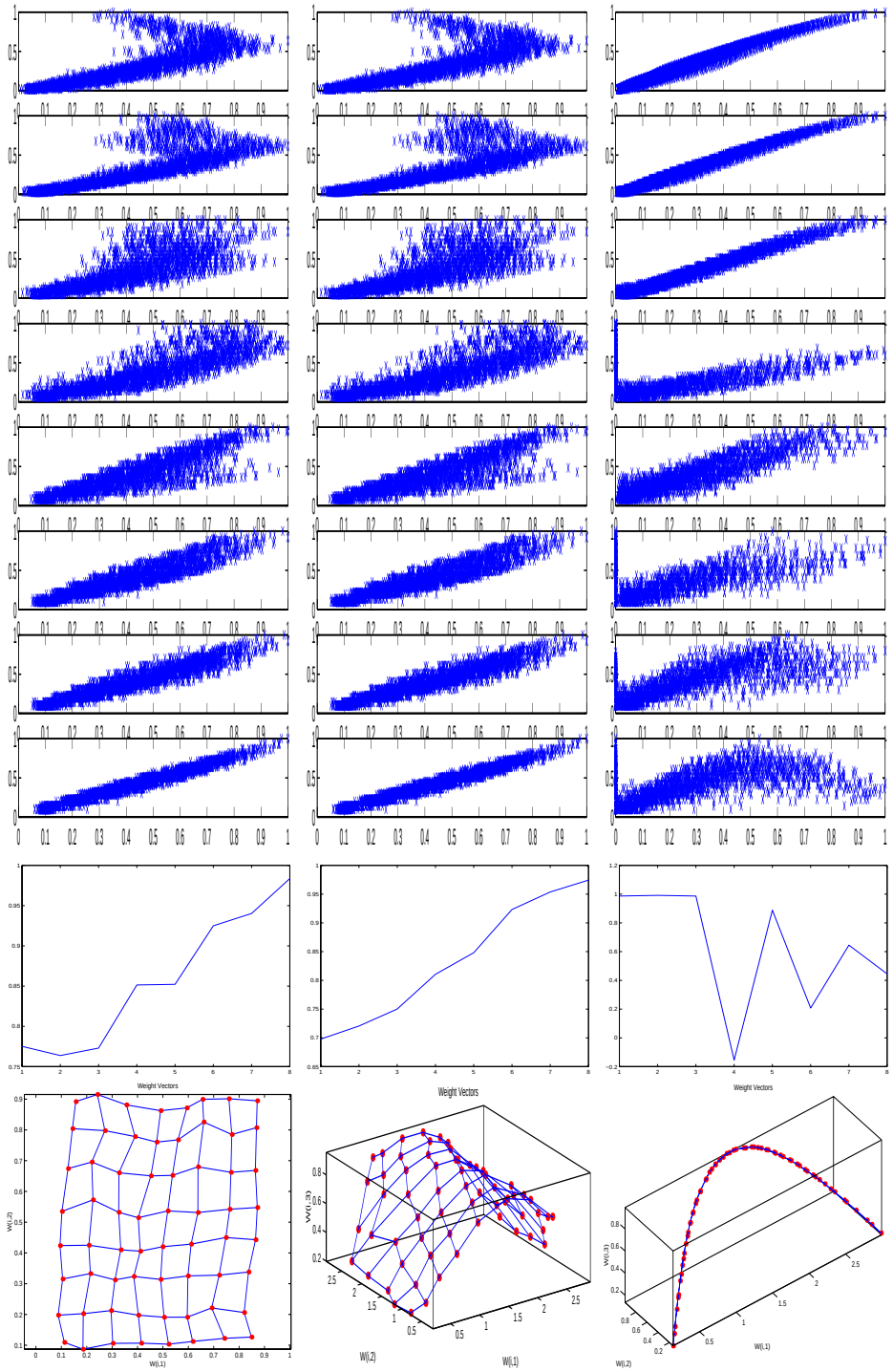


Fig. 4.3: exemples A, B et C

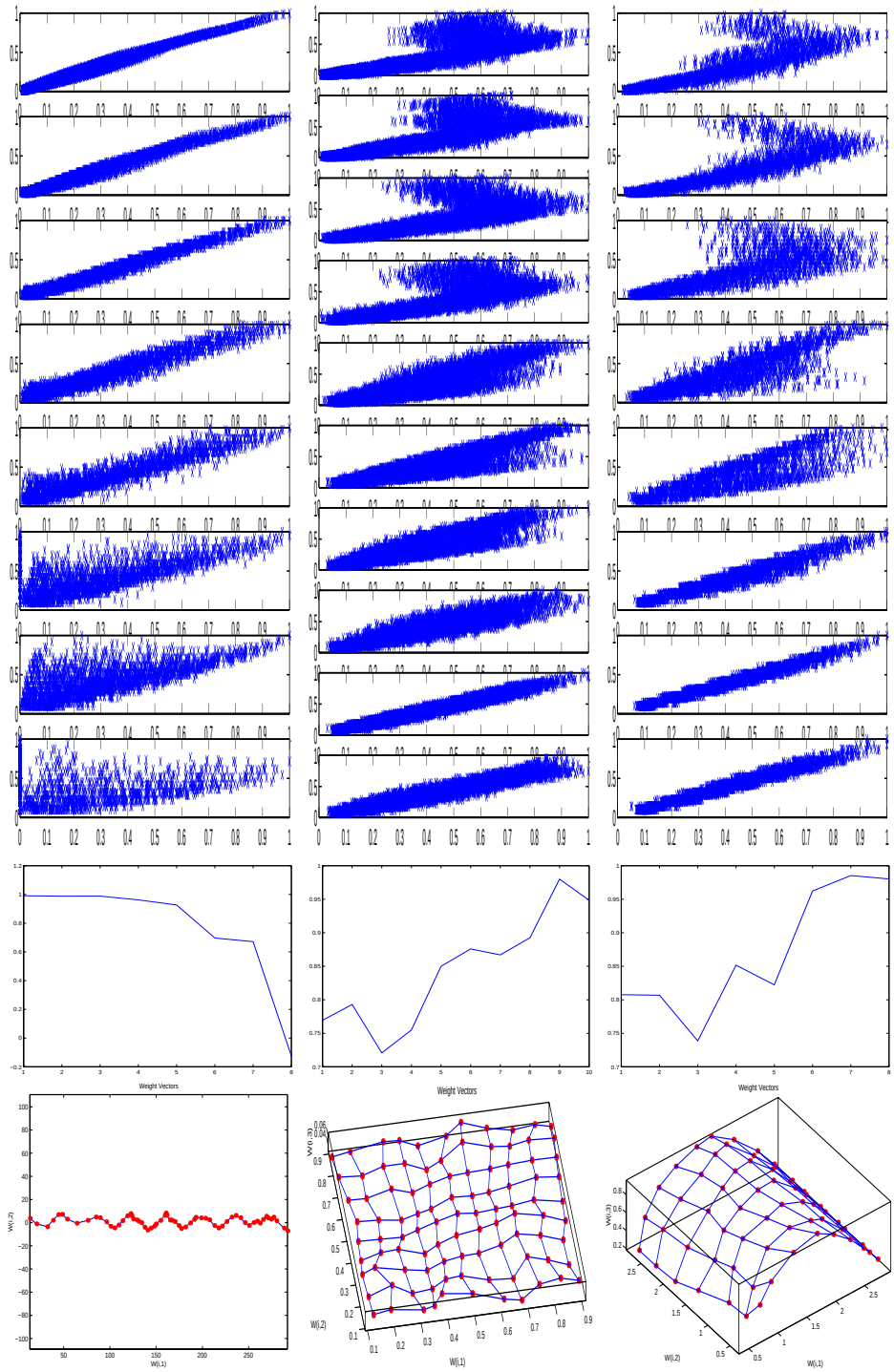


Fig. 4.4: exemples D, A' et B'

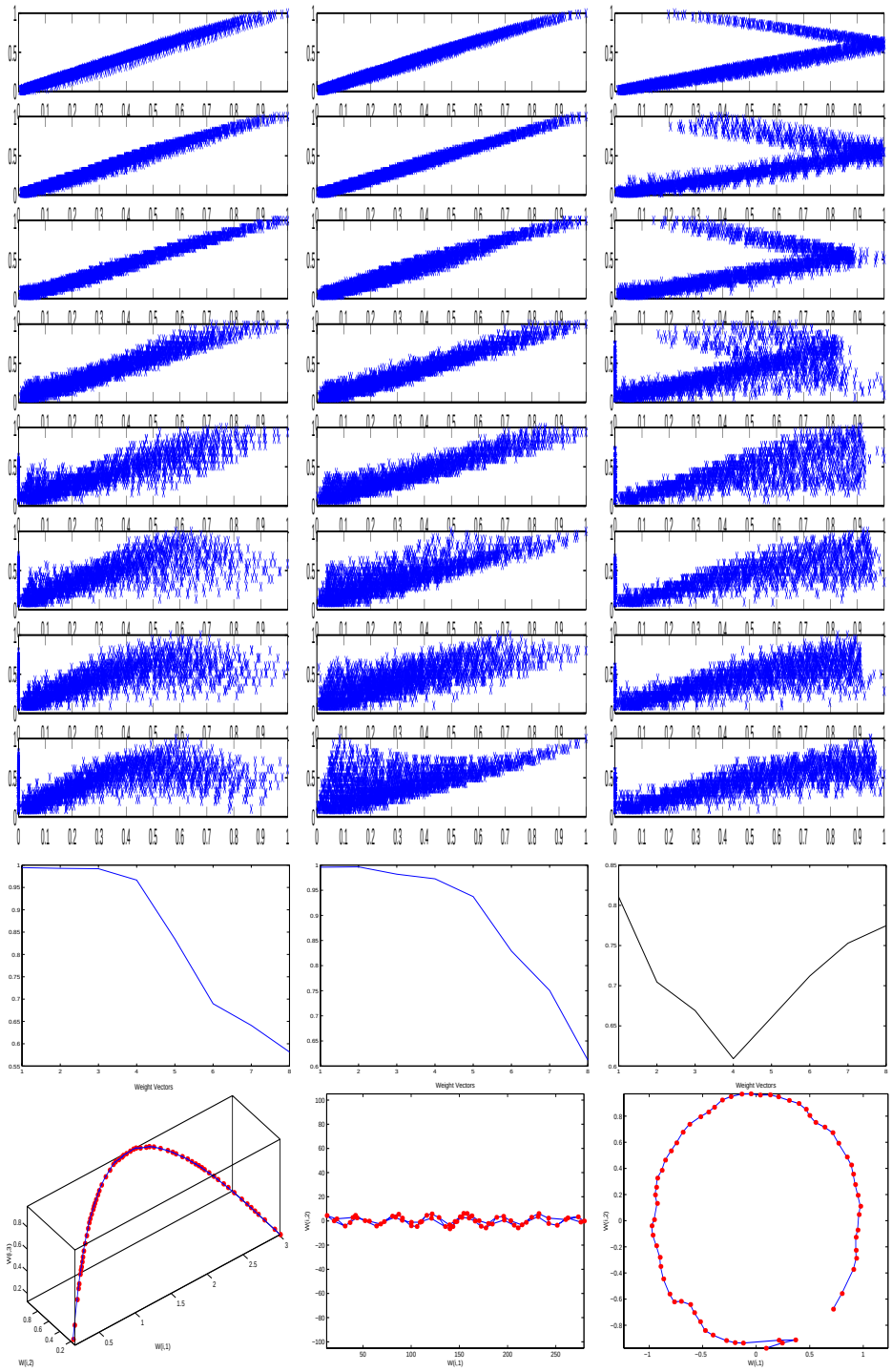


Fig. 4.5: exemples C', D' et E

Détection d'un effet sur-apprentissage

Même en paramètrant une carte de Kohonen "parfaitement" c'est-à-dire en connaissant la structure (dimension et rapport de longueur entre les axes), un trop grand nombre de cellules demandées en entrée de la carte de Kohonen peut induire un phénomène de sur-apprentissage avec une mauvaise représentation de la topologie des données.

Les exemples suivants (un tirage uniforme de 100 points sur $[0,1]^2$ et de 200 points sur une sinusoïde) montrent cet effet : on a appliqué des cartes de Kohonen (carrées et linéaires) avec un nombre d'unités croissant (2^2, 3^2...;10^2 pour le premier tirage et 10, 20,...160 pour le second) et on a observé l'évolution de la corrélation entre les deux distances en fonction du nombre d'unités.

Dans le second cas, le fait que l'algorithme soit stochastique induit des discontinuités dans l'évolution de la corrélation, on a calculé trois cartes par exemples.

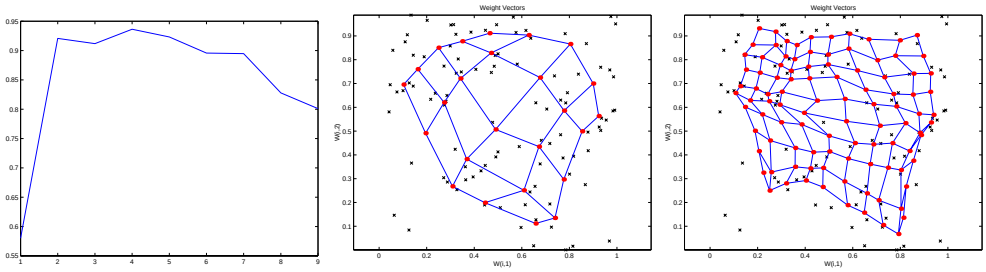


Fig. 4.6: 100 points tirés uniformément sur un carré, évolution du coefficient de préservation de la topologie en fonction du nombre de cellules dans la carte, carte 5×5 (maximum de correlation) et carte 10×10 (sur-apprentissage)

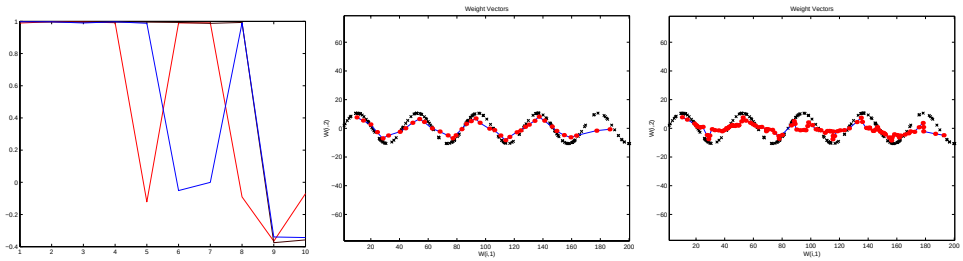


Fig. 4.7: 200 points tirés sur une sinusoïde,évolution du coefficient de préservation de la topologie en fonction du nombre de cellules dans la carte (sur 3 essais de ficelles), ficelles de taille 60 et 160.

Evolution de la corrélation entre les distances au cours de l'algorithme

Il apparaît qu'au cours de l'apprentissage de la carte, la corrélation entre les deux distance est croissante en moyenne. Pour éviter un temps de calcul trop long, on a ici "triché" sur les résultats en utilisant notre connaissance a priori des tirages pour ne pas calculer étape par étape la distance curviligne entre les vecteurs codes.

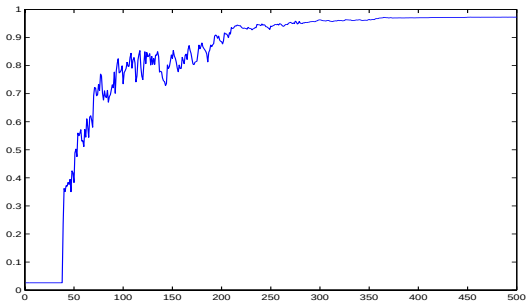


Fig. 4.8: évolution de la correlation entre les distances au cours de l'algorithme pour un tirage carré.

4.2.4 Consolidation de la dimension

Un des problèmes non évoqué encore dans ce chapitre sur l'estimation de dimension, présent surtout dans le cas des méthodes de types "correlation dimension" ou "capacity dimension" pour lesquelles on doit choisir un endroit pour calculer une pente, est le problème de la mesure de la dimension du bruit. Il est illustré sur la figure suivante qui représente visiblement un ensemble de dimension 1 bruité, de manière à ce que la valeur du seuil de connexité soit de l'ordre des rayons pour lesquels la dimension de correlation donnera 2 :

Dans ce cas de figure, la validation du paramétrage d'une carte de Kohonen sera une aide majeure à l'estimation de la dimension.

Revenons sur la Box Counting dimension, qu'on n'avait pas détaillée dans le chapitre d'estimation de dimension du fait de la difficulté de sa mise en oeuvre. Le principe était de calculer $N(\epsilon)$ nombre de "boîtes" de coté (ϵ) nécessaires au recouvrement des données. On calculait alors la dimension comme :

$$D_{cap} = \lim_{\epsilon \to 0} - \frac{log(N(\epsilon))}{log(\epsilon)}$$

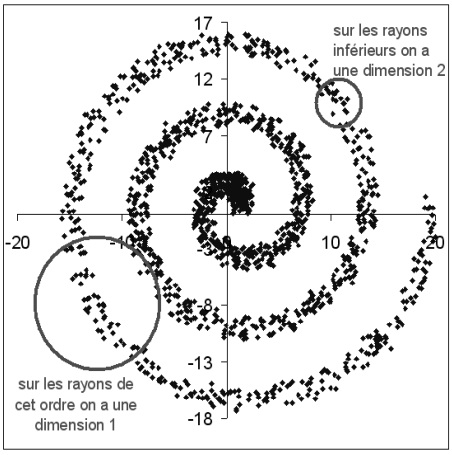


Fig. 4.9: Illustration du problème de mesure du bruit en "correlation" dimension.

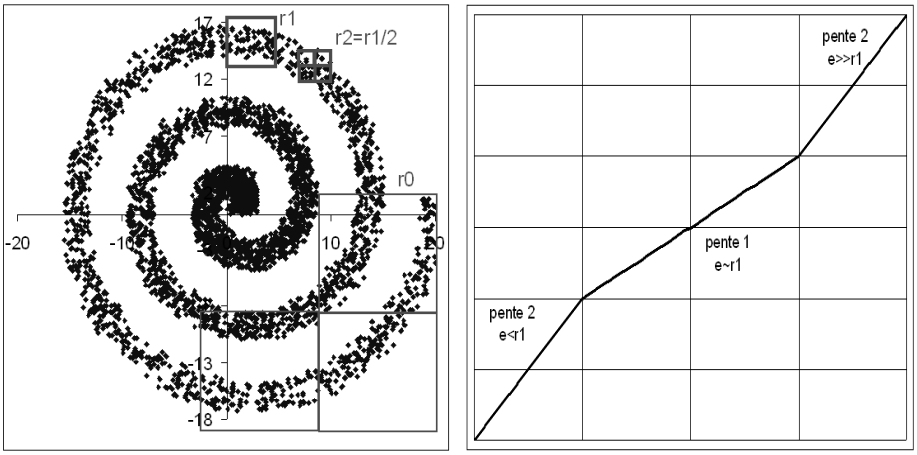


Fig. 4.10: Et le même problème en "box counting dimension".

Soit alors un ensemble homéomorphe à $[0, L_1] \times [0, L_2] \times ... \times [0, L_d]$ avec $L_1 \geq L_2 \geq .... \geq L_d$. Pour des tailles de $\epsilon < L_d$, on observera sur le graphique une pente de d. Or l'information donnée par les longueurs est elle aussi fondamentale. Dans l'exemple précédent, on observait " à l'oeil" une dimension 1 du fait que $L_1 >> L_2$. Les longueurs relatives dans toutes les dimensions peuvent se déduire de la paramétrisation des cartes de Kohonen explicitée dans le chapitre. Ainsi, par exemple, si la dimension estimée par une méthode quelconque d'estimation de dimension est de 2, on va tester un ensemble de cartes de Kohonen allant de

la ficelle à la carte "carrée" et si, par exemple encore, la meilleure paramétrisation est $(50, 2)$ on pourra en déduire que l'espace des données est de dimension 1 avec du bruit.

4.3 Résultats et retour au test de connexité

4.3.1 Méthodologie

On commence par estimer la dimension des observations en utilisant une des méthodes d'estimation de dimension issues de la théorie fractale (dans les exemples suivants on a choisi la "correlation dimension"). Une fois la dimension maximum calculée, on teste un ensemble de cartes de Kohonen de dimension inférieure ou égale. Dans les exemples suivants, la dimension estimée est 2 et on teste des cartes de Kohonen d'environ 100 cellules. On part d'une carte $(100, 1)$, c'est-à-dire de dimension 1 pour aboutir à une carte $(10, 10)$ de dimension 2. On choisit la meilleure paramétrisation au sens du critère de corrélation entre la distance curviligne dans les données et la distance euclidienne dans la carte. On estime les longueurs L_i dans les différentes dimensions de la manière suivante pour une carte rectangulaire (les calculs étant parfaitement analogues dans des dimensions supérieures) :

$$L_1^*(i) = \sum_{j=1}^{n_1} ||C_{j,i} - C_{j+1,i}||_2^2$$

où $C_{i,j}$ correspond au vecteur code (i, j)

$$L_1' = \frac{1}{n_1} L_1(i)$$

$$L_1 = \frac{n_1 + 2}{n_1} L_1'.$$

Cela signifie qu'on regarde la longueur moyenne de la carte suivant la première dimension, que l'on pondére en fonction du nombre de cellules pour prendre en compte le fait que les vecteurs codes représentent les centres des cellules et non leurs extrémités.

L'algorithme suivant résume la méthode finale adoptée :

- Estimation de la dimension

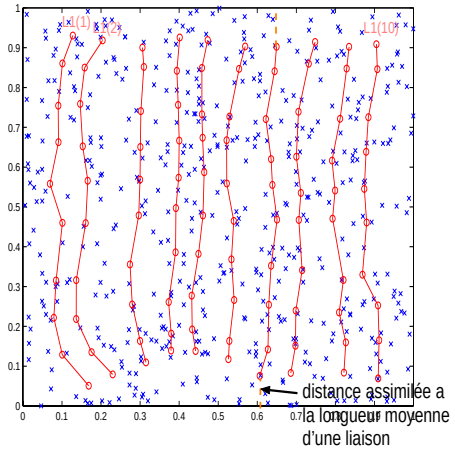


Fig. 4.11: Illustration du calcul de la longueur dans la direction "verticale"

- Recherche de la "meilleure" carte de Kohonen dans cette dimension (et les dimensions inférieures)
- Calcul des "longueurs" dans chacune des directions de la carte ce qui nous permettra d'affiner la dimension
- Test de connexité

4.3.2 Quelques résultats

On a simulé des données de la manière suivante :

- $X(:,1)$ suit une loi uniforme sur $[0,30]$
- $X(:,2)$ suit une loi uniforme sur $[0,1]$
- $X(:,1) = sin(X(:,1)/2)$

On présente les graphiques suivants pour illustrer les résultats :

- figure 5.12 : les données
- figure 5.13 : l'estimation de la dimension intrinsèque par la dimension de corrélation donne 2.
- figure 5.14 : la "meilleure" carte de Kohonen en dimension 2 pour environ 100 cellules est la carte $(3,33)$ et l'estimation des longueurs nous montre que la longueur dans la première dimension

est négligeable devant la longueur dans la deuxième dimension, les données sont ainsi en dimension 1 et perturbées par un bruit.

- figure 5.15 : la lecture graphique du test de connexité (par rapport à un tirage uniforme dans $[0, L_1] \times [0, L_2]$) nous indique une seule composante connexe
- figure 5.16 : à titre indicatif en testant la connexité par rapport à un tirage uniforme dans $[0, 30] \times [0, 1] \times [-1, 1]$, on a choisi deux composantes connexes

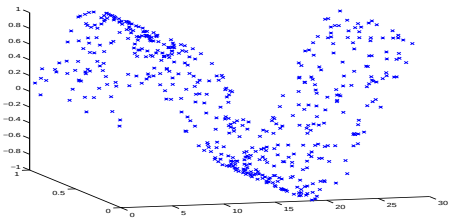


Fig. 4.12: observation.

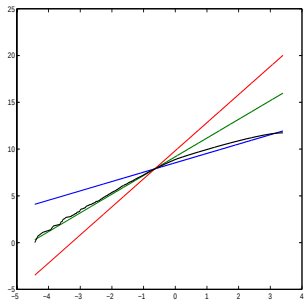


Fig. 4.13: On estime la dimension a 2

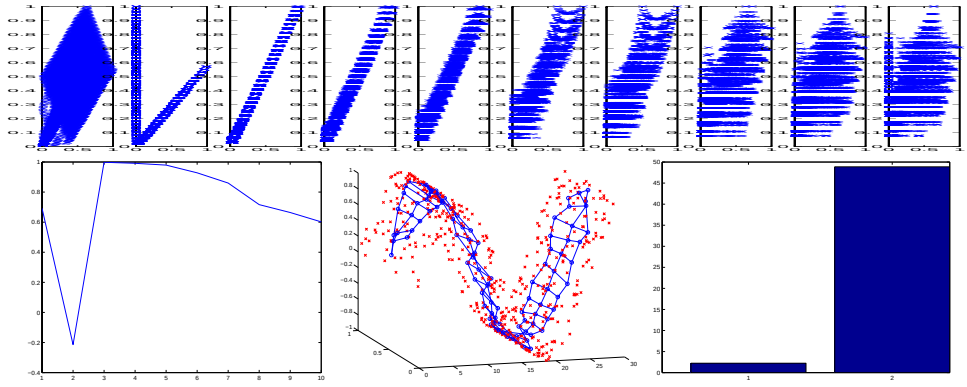


Fig. 4.14: L'étude de différentes paramétrisation des cartes de Kohonen indique que la dimension est essentiellement 1

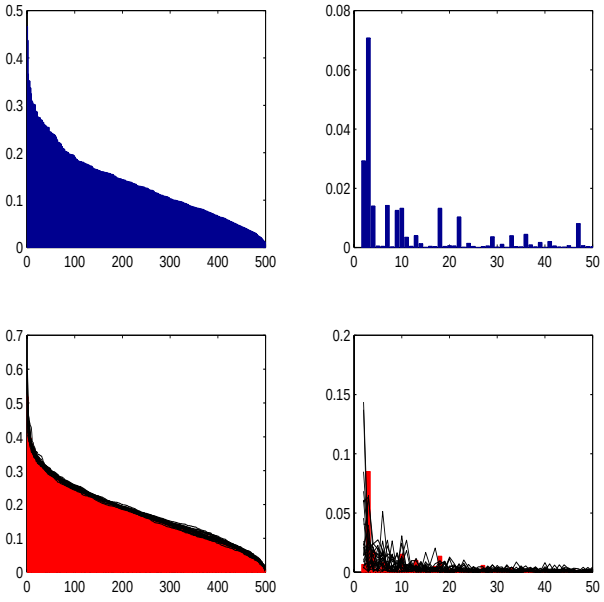


Fig. 4.15: L'ensemble est constitué d'une seule classe connexe

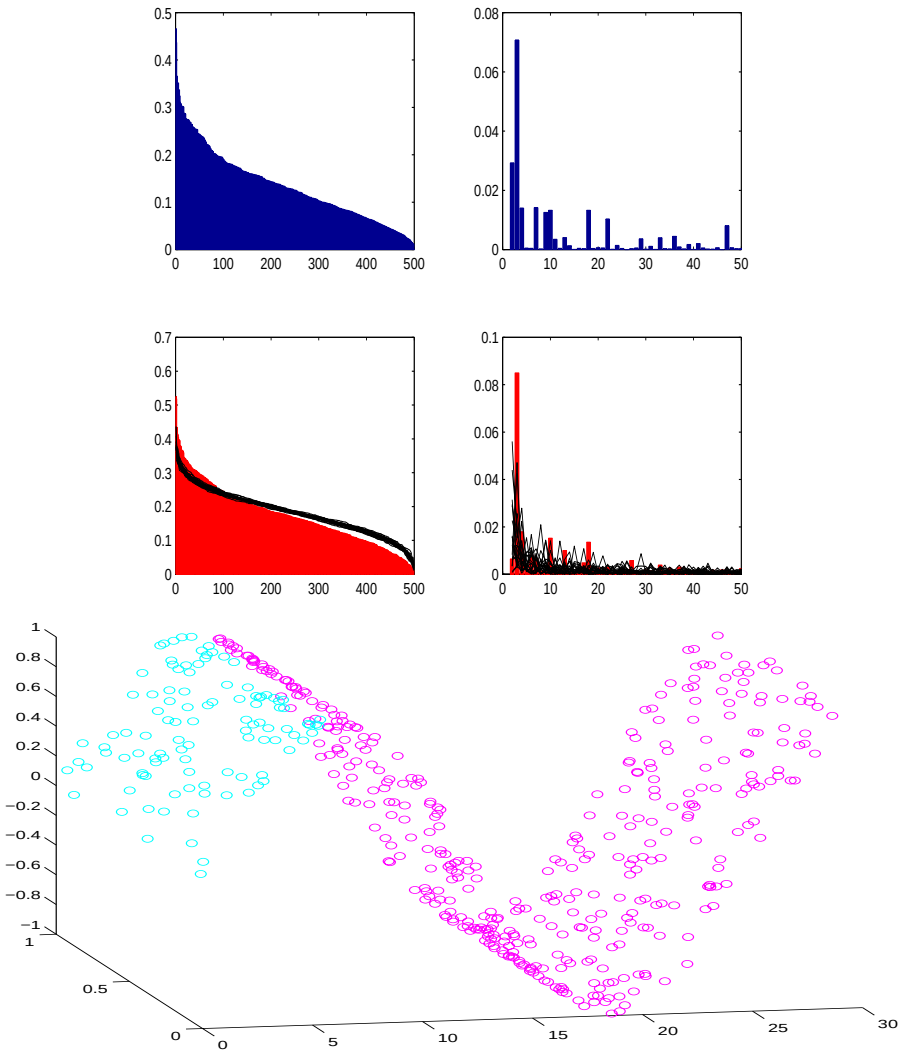


Fig. 4.16: En testant la connexité dans un ensemble parallélépipèdique on obtiendrait deux composantes connexes

Un deuxième exemple, similaire, en changeant la fréquence de la sinusoïde donne les mêmes types de résultats : On a simulé des données de la manière suivante :

- $X(:,1)$ suit une loi uniforme sur $[0,30]$
- $X(:,2)$ suit une loi uniforme sur $[0,1]$
- $X(:,1) = sin(X(:,1)/2)$

On présente les graphiques suivants pour illustrer les résultats :

- figure 5.17 : les données
- figure 5.18 : l'estimation de la dimension intrinsèque par la dimension de corrélation nous donne 2
- figure 5.19 : la "meilleure" carte de Kohonen en dimension 2 pour environ 100 cellules est la carte $(3,33)$ et l'estimation des longueurs nous montre que la longueur dans la première dimension est négligeable devant la longueur dans la deuxième dimension, les données sont ainsi en dimension 1 avec un bruit.
- figure 5.20 : la lecture graphique du test de connexité (par rapport à un tirage uniforme dans $[0,L_1] \times [0,L_2]$) nous indique une seule composante connexe
- figure 5.21 : à titre indicatif en testant la connexité par rapport a un tirage uniforme dans $[0,30] \times [0,1] \times [-1,1]$ on aurait choisi trois composantes connexes

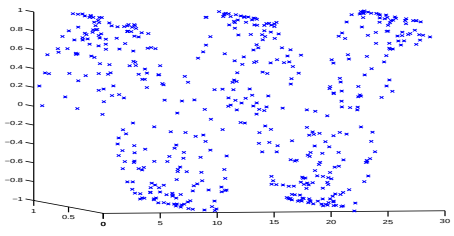


Fig. 4.17: Observation.

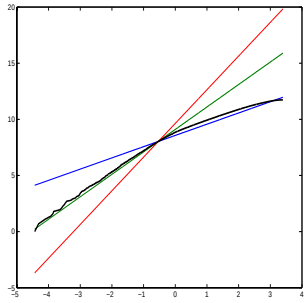


Fig. 4.18: On estime la dimension a 2

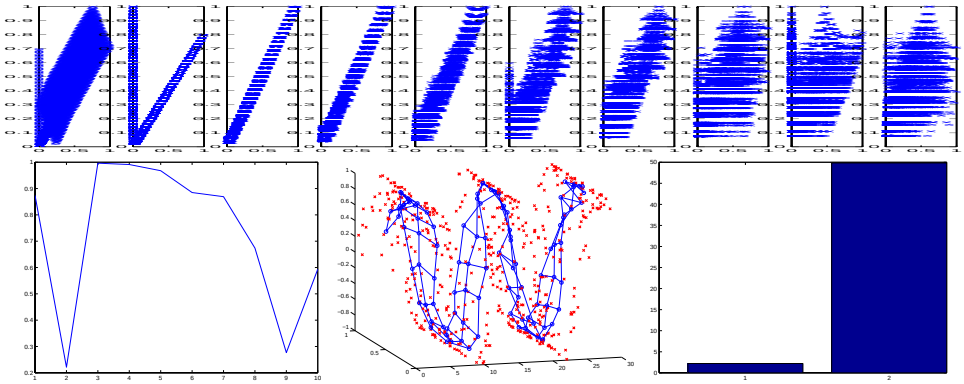


Fig. 4.19: L'étude de différentes paramétrisation des cartes de Kohonen indique que la dimension est essentiellement 1

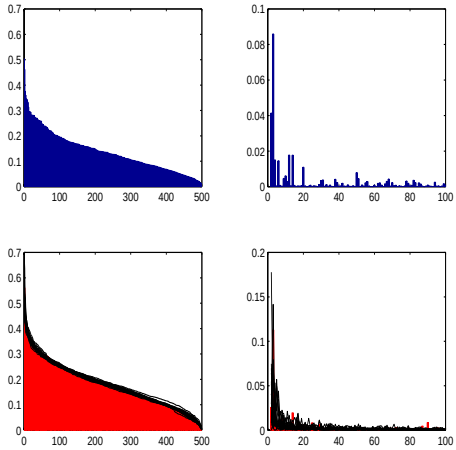


Fig. 4.20: L'ensemble est constitué d'une seule classe connexe

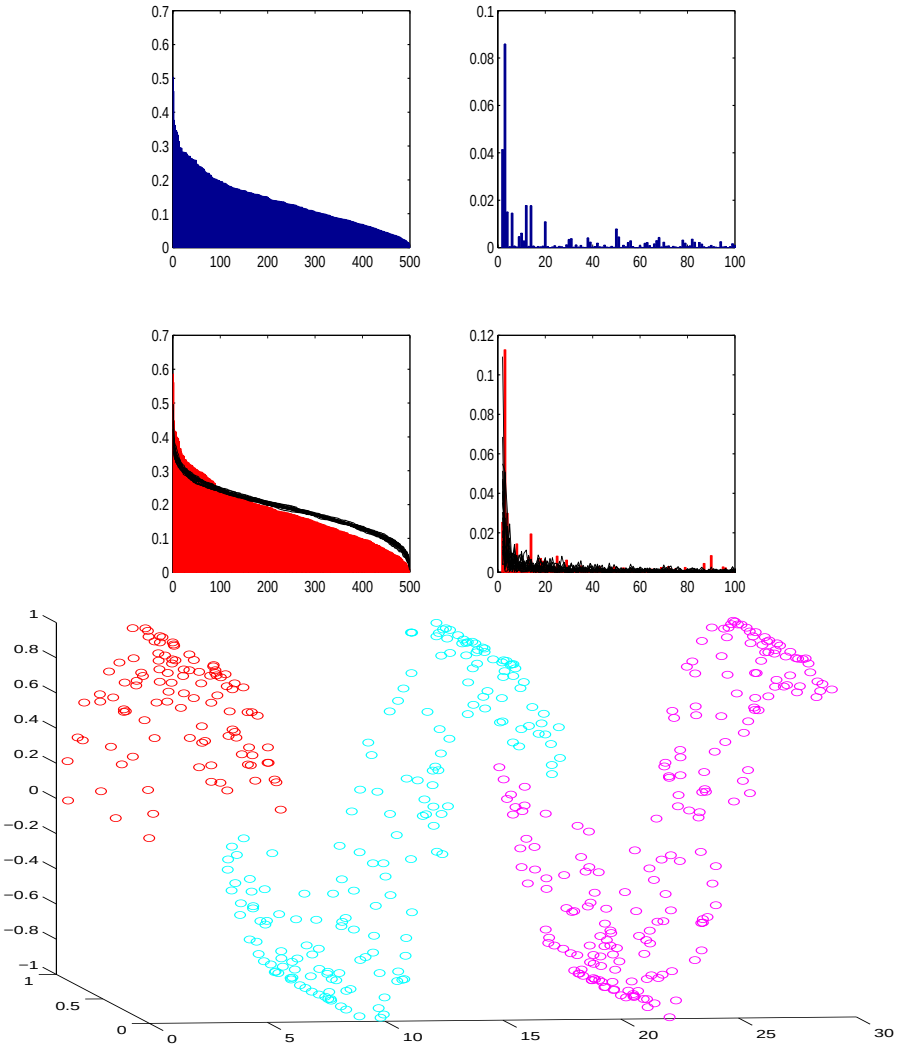


Fig. 4.21: En testant la connexité dans un ensemble parallélépipèdique on obtiendrait deux composantes connexes

5. CONCLUSION ET PERSPECTIVES

D'une part il faut noter que les résultats de la normalisation vont au delà de nos espérances, mais que l'aspect théorique nous échappe. Dans le cas de tirages uniformes sur un rectangle, il est clair que ce sont les "problèmes" de bord qui permettent d'obtenir au final une normalisation "carrée". Dans le cas général, les phénomènes impliqués sont plus obscurs et difficile à mettre en équations. La recherche de la "meilleure" paramétrisation d'une carte de Kohonen offre de nombreuses perspectives en analyse des classes et surtout en connaissance de la dimension. Tout a été fait pour que l'adaptation des algorithmes à des dimensions supérieures à 2 soit aisée, mais ce travail n'est pas terminé. Pour pouvoir en tester les résultats sur des exemples autres que "jouet" mais aussi avec des données réelles, il faudrait au préalable trouver un algorithme convergeant vers la "meilleure" carte pour éviter de toutes les tester, ce qui donne un temps de calcul trop élevé. Nous étudions actuellement quelques idées allant dans ce sens.

Une perspective à l'ensemble des deux parties (classification et analyse d'une classe sous hypothèse de connexité) est la mise en place d'indicateurs préliminaires à une modélisation. En effet supposons qu'on recherche un modèle de la forme $Y = f(X)$ avec f fonction continue. Alors pour des problèmes de raccordement, il faudra travailler sur des composantes connexes par arc en X. Supposons que X soit connexe par arc (le cas échéant on travaille par composante connexe de X) alors si $Y = f(X)$ avec f continue le graphe, $\{(x, f(x)\}$ est connexe par arc (la réciproque est fausse mais garantit quand même que l'ensemble des points de discontinuité de f ne sépare pas l'ensemble de départ en plusieurs classes connexes par arc). Ainsi, si X est connexe et (X, Y) n'est pas connexe, la recherche d'un modèle sera vaine. Supposons maintenant que X et (X, Y) soient connexes, si on a un modèle $Y = f(X)$, alors la dimension intrinsèque de X et la dimension intrinsèque de (X, Y) sont identiques. Bien sûr il faut pouvoir envisager un modèle $Y = f(X) + \varepsilon$

mais dans ce cas on peut imaginer que si les mesures de dimension intrinsèque ne sont alors pas identiques pour X et (X, Y), le nombre de "longueurs" significatives est le même. L'estimation des longueurs dans les différentes direction pourrait alors nous aider à construire un coefficient de corrélation non linéaire. Contrairement au coefficient de correlation non linéaire de Spearman, ce dernier ne nécessite pas d'hypothèse sur la monotonie de la fonction du modèle et, de plus, serait calculable pour des données de dimension quelconque.

Part III

APPLICATION DES CARTES DE KOHONEN A DES DONNÉES TEMPORELLES ÉCONOMIQUES

Ces deux derniers travaux ont été réalisés en collaboration étroite avec des économistes. Dans le premier cas (étude des dynamiques individuelles), le travail a été effectué avec Corinne Perraudin et Joseph Rinkievicz dans le cadre du développement des cartes de Kohonen au SAMOS. Le second (étude des dynamiques de groupes) a été avec fait avec Yamina Tadjeddine et Sebastien Galanti dans le cadre d'une étude commandée par la CDC (Caisse des Dépôts et Consignations) sur l'analyse du comportement des gérants de fonds.

1. ETUDE DES DYNAMIQUES INDIVIDUELLES

1.1 Introduction

L'Union Economique et Monétaire (UEM) est une étape logique dans la construction européenne, mais constitue en même temps un réel changement politique pour les pays y participant. En effet, l'introduction d'une monnaie européenne unique gérée par une Banque Centrale Européenne correspond à l'abandon d'un fort instrument de souveraineté. Le traité signé à Maastricht, le 7 février 1992, rend irréversible la marche vers une monnaie unique en définissant un calendrier des réalisations à effectuer en trois phases.

Durant la première phase, les états membres s'engagent à présenter des "programmes de convergence" visant à rapprocher et à améliorer leurs performances économiques afin de rendre possible l'établissement de parités fixes entre leurs monnaies.

La deuxième phase de l'UEM, qui débute le 1er janvier 1994, constitue une période transitoire, au cours de laquelle les efforts de convergence sont maintenus et amplifiés. L'Institut Monétaire Européen (IME) est mis en place à Francfort; ses missions sont de renforcer la coordination des politiques monétaires, de promouvoir le rôle de l'Euro et de préparer l'installation de la Banque Centrale Européenne pour la troisième étape.

En mai 1998, les ministres des Finances des Quinze établissent, sur la base de rapports établis par la Commission et l'IME, une liste des États membres qui remplissent les conditions leur permettant de passer à la monnaie unique, ce qui constitue la troisième phase.

Les normes fixées par le traité de Maastricht pour une éventuelle participation à l'Union Economique et Monétaire sont exclusivement d'ordre monétaire et financier. Elles visent à rapprocher les comportements des pays en matière d'inflation, de taux d'intérêt, de déficit budgétaire, de dette publique et de taux de change[1]. Pour l'essentiel,

[1] Les critères de Maastricht sont les suivants : (1) le taux d'inflation ne peut dépasser de plus de 1,5% la moyenne des taux des trois États ayant la plus faible

il s'agit donc d'assurer la convergence nominale des économies des pays-membres. L'hypothèse sous-jacente est que la stabilité des taux de change et des prix favorisera la croissance et l'intégration économique, de sorte que les pays qui cherchent à atteindre des cibles nominales communes verront également converger leur structure économique. De plus, l'existence d'une monnaie unique et d'une politique monétaire commune, en contraignant les pays membres de l'union monétaire à une réponse commune et uniforme, requière une certaine intégration nominale.

Sur la base de leurs performances économiques en 1998, 11 pays ont formé l'UEM et ont adopté l'Euro comme monnaie : l'Allemagne, l'Autriche, la Belgique, l'Espagne, la Finlande, la France, l'Irlande, l'Italie, le Luxembourg, les Pays-Bas et le Portugal. La Grèce a intégré la zone Euro en 2001. Le Danemark et le Royaume-Uni n'ont pas souhaité intégrer la zone Euro et la Suède intégrera l'UEM dès qu'elle aura rempli les conditions.

L'objectif de cet article est d'étudier la convergence nominale des économies des pays actuellement dans l'Union Européenne, sans a priori théorique, à savoir en étudiant la plus ou moins grande homogénéité des économies sur la base de données relatives aux critères de Maastricht. Plus précisément, l'objectif est d'étudier la transition de ces économies vers les critères définis par le traité de Maastricht pour participer à l'UEM. Il est en effet intéressant d'étudier les processus de convergence des économies européennes et de voir s'il existe une certaine homogénéité dans les processus de transition vers les normes communes ou si l'on peut observer l'idée d'Europe à plusieurs vitesses, y compris dans les transitions.

Nous considérons dans cette étude les 15 pays actuellement dans l'Union Européenne, même si seulement 12 d'entre eux font partie de l'UEM aujourd'hui. L'étude est basée sur la période 1980-2002. Nous utilisons donc des données temporelles relatives à 4 variables économiques pour 15 pays[2] : le déficit public en pourcentage du PIB, la dette publique en pourcentage du PIB, le taux d'inflation (taux de croissance de l'indice des prix à la consommation harmonisé) et le taux d'intérêt nominal de

inflation; (2) les taux d'intérêt à long terme ne peuvent varier de plus de 2% par rapport à la moyenne des taux des trois États les plus bas; (3) les déficits budgétaires nationaux doivent être proches ou inférieurs à 3% du PNB; (4) la dette publique ne peut excéder 60% du PNB que si elle a tendance à descendre vers ce niveau; (5) une monnaie nationale ne peut avoir été dévaluée au cours des deux années précédentes et doit être restée dans la marge de fluctuation de 2,25% prévue par le SME.

[2] Les données proviennent de Economic Outlook de l'OCDE.

long-terme[3].

Afin d'étudier les trajectoires des pays européens et de définir des classes de pays européens, nous proposons d'adapter l'algorithme de Kohonen[4] (encore appelé algorithme SOM) au traitement de données temporelles et spatiales. Les sections suivantes présentent la méthode et les résultats obtenus.

1.2 Une carte de Kohonen contrainte

1.2.1 Le principe

Afin de prendre en compte la dimension temporelle des données, on pourrait construire autant de cartes de Kohonen que d'années et classifier les 15 observations (pays) selon les 4 variables retenues par année[5] pour des applications directes de l'algorithme de Kohonen sur des données temporelles, traitant de transition d'individus sur la carte en ayant considéré comme observation un individu pour une année.. Le problème avec ce type de méthode est que la classification ainsi obtenue année par année est très instable. Ainsi, nous proposons une méthode qui prend en compte simultanément l'ordonnancement temporel et spatial des données. Pour cela, on construit une carte de longueur 23 (nombre d'années) et de largeur 8 (nombre de représentants par année choisi a priori), pour laquelle le calcul des vecteurs codes s'effectue selon l'algorithme suivant :

- L'initialisation de l'algorithme SOM correspond à un tirage aléatoire de 8 pays dans l'ensemble des données. A l'unité (i, t) de la carte, on affecte les quatre valeurs des variables[6] du pays i pour l'année t.

- A chaque itération, un pays i_0 et une année t_0 sont tirés aléatoirement dans l'ensemble de données. Ensuite, pour tout $i \in [1, 8]$, on cherche l'unité (i, t_0) qui est la plus proche de l'observation sélectionnée.

[3] Nous ne retenons pas le taux de change parmi nos variables, car le critère qui y réfère repose sur une stabilité des changes et non pas sur une norme donnée.

[4] Nous supposons que le lecteur est familier avec l'algorithme de Kohonen. Voir par exemple [DYI3] ou [DYI4] pour une présentation de l'algorithme et pour des applications à l'analyse de données diverses.

[5] Voir [DYI1] ou [DYI2]

[6] Les données sont centrées et réduites par variable sur la période entière.

- On met à jour l'unité gagnante et les unités voisines. Le voisinage décroît dans la dimension ligne durant les itérations de r à 0. Pour forcer l'organisation temporelle, pour un voisinage ligne donné r, la taille du voisinage temporel décroît de r à 0 (voir figure 1.2.B).

- Finalement, afin de garantir la convergence, on finit à 0 voisin sur les deux derniers tiers des itérations.

Une fois que l'algorithme a convergé, on place les pays sur la carte afin d'identifier leur position.

1.2.2 La classification

La carte ainsi obtenue permet d'observer une continuité à la fois dans la dimension temporelle et dans la dimension ligne (voir figure 1.1). Une opposition très nette apparaît sur la carte entre le coté en haut à droite, qui correspond à des niveaux élevés des 4 variables (dépassant les normes de Maastricht), et le bas à gauche de la carte, qui correspond à des niveaux faibles des 4 variables. Pour chaque ligne (soit pour chaque année), les performances en termes de normes de Maastricht décroissent quand on parcourt les unités de la gauche vers la droite. On peut observer un resserrement des profils moyens extrêmes par année durant la période 1980-2002. La différence entre les meilleures et les moins bonnes performances diminue au cours des années. Ces résultats illustrent le processus de convergence des pays européens vers les critères de Maastricht.

Le nombre de classes est ensuite réduit par une classification hiérarchique appliquée aux 8×23 vecteurs codes. Le nombre de classes retenu est 5 (voir figure 1.1 où les classes sont indiquées par une échelle de gris et sont délimitées par une ligne noire).

Premièrement, on peut observer que les classes regroupent des unités correspondant à plusieurs années et que le nombre de classes par année varie entre 1980 et 2002. Chaque année, de 1980 jusqu'à 1994, les 15 pays sont regroupés dans 3 classes (sauf 1985, qui est caractérisé par 4 classes, et 1986 dont une des 4 classes ne contient pas d'observation). A partir de 1995, il ne reste que 2 classes. La diminution du nombre de classes durant la période étudiée illustre à nouveau le processus de convergence des pays européens.

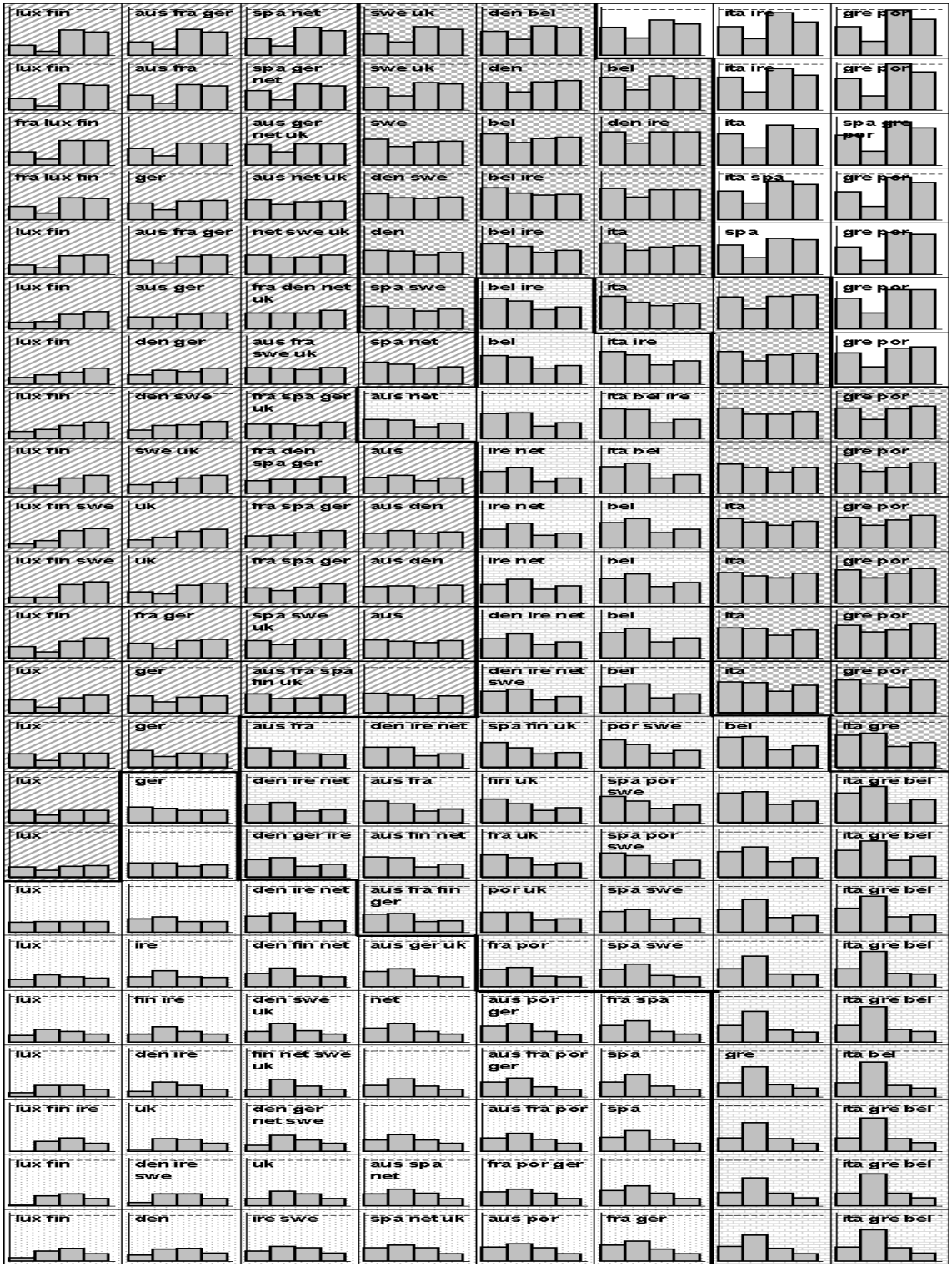


Fig. 1.1: Carte de Kohonen contrainte

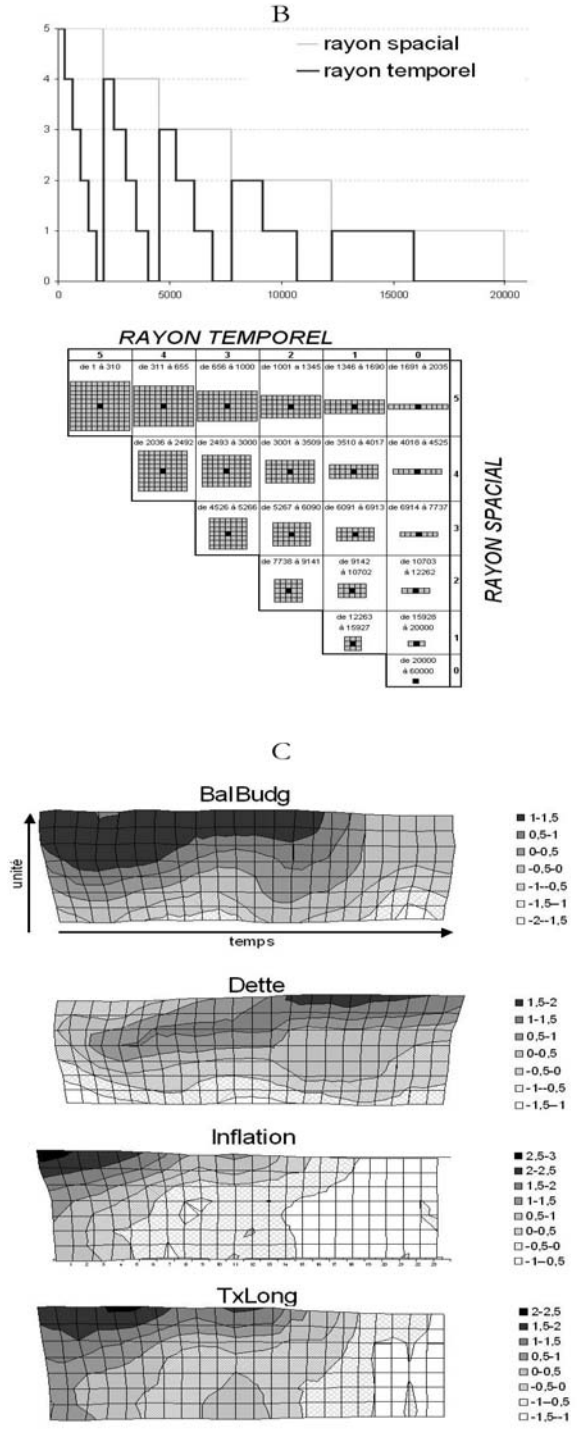


Fig. 1.2: B : Evolution du rayon dans l'algorithme modifié; C : Projection des variables sur la carte transposée.

Deuxièmement, on observe que les classes se déplacent vers le coté droit de la carte de Kohonen, indiquant que les mauvais profils d'une année correspondaient avant aux profils moyens. Par exemple, la classe regroupant l'Italie, l'Irlande, la Grèce et le Portugal en 1980-81 disparaît en 1987; le profil de l'Italie, de la Grèce et du Portugal en 1990 se retrouve dans la même classe que celui de la Suède, du Royaume-Uni, du Danemark et de la Belgique en 1980. Ainsi, la différence dans les performances entre les pays existant dans les années 80 a progressivement disparu et on peut remarquer une plus forte homogénéité entre les pays européens à la fin de la période sur la base des critères de Maastricht.

Plus précisément, au début des années 80, la classe la plus vertueuse selon les critères de Maastricht est composée du Luxembourg, de l'Autriche, de la France, de l'Allemagne, de l'Espagne et des Pays-Bas. Cette classe regroupe les pays qui constituaient le cœur de la Communauté Economique Européenne, ainsi que l'Espagne et l'Autriche. La classe intermédiaire est composée de la Suède, du Royaume-Uni, du Danemark et de la Belgique. Et la dernière classe est composée de l'Italie, de l'Irlande, de la Grèce et du Portugal. Nous pouvons remarquer que l'Italie, la troisième puissance européenne, reste dans le classe des pays les moins vertueux, révélant les difficultés à équilibrer ses finances publiques et à ralentir son inflation.

Un changement important apparaît au début des années 90 avec le commencement d'une profonde récession et avec la crise du Système Monétaire Européen (SME). Plus précisément, en 1993, la classe intermédiaire devient plus importante, regroupant des pays qui appartenaient précédemment à la classe la plus vertueuse. En 1994 et 1995, la classe la plus vertueuse n'est composée que du Luxembourg. Cela illustre les effets de la récession en Europe, qui a provoqué d'importants déficits publics et la fin du SME. Après 1995, il ne reste que deux classes. La classe la plus vertueuse regroupe progressivement de plus en plus de pays, puisqu'en 1997-1998 les pays membres de l'UE se doivent de respecter les critères de Maastricht pour intégrer l'UEM. En 2002, trois pays restent dans la classe la moins vertueuse (Italie, Grèce et Belgique) puisqu'ils continuent à avoir un ratio de dette publique sur PIB qui dépasse les 100%, et donc qui dépasse la norme fixée à 60%. Cependant, puisqu'ils ont réussi à faire décroître suffisamment ce ratio, cette mauvaise performance n'a pas été un obstacle à leur intégration à l'UEM.

On peut remarquer que la France et l'Allemagne en 2002 sont très proches de la classe la moins vertueuse. Comme chacun s'en souvient, les deux grandes puissances européennes ont joué le rôle de mauvais élèves en 2002, à cause des problèmes qu'ils ont eu à équilibrer leurs finances publiques malgré leur engagement dans le cadre du Pacte de Stabilité. En effet, depuis 1997, le Pacte de Stabilité et de croissance limite la possibilité pour un pays membre de trop user de politiques fiscales expansionnistes pour faire face à une récession. Le non respect de la norme sur les finances publiques a alors conduit la Commission Européenne à déclencher la procédure de déficit excessif pour l'Allemagne et la France en 2002, comme elle l'avait déjà fait pour le Portugal en 2001 (qui se retrouve proche de la classe la moins vertueuse en 2001).

1.2.3 Trajectoires individuelles

Nous pouvons représenter les trajectoires de chaque pays à travers la carte. La trajectoire du Luxembourg (voir figure 1.3) est très proche du coté gauche de la carte de Kohonen, indiquant qu'il a été le plus vertueux puisqu'il a été caractérisé sur toute la période par les valeurs les plus faibles pour les 4 critères.

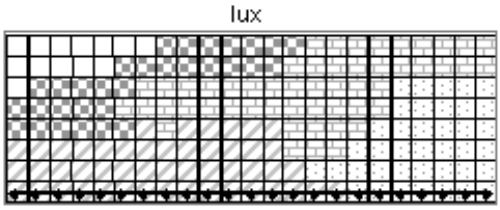


Fig. 1.3: Trajectoires du Luxembourg

Les trajectoires de l'Autriche, de la France, de l'Allemagne, des Pays-Bas, du Royaume-Uni, du Danemark, de la Finlande et de la Suède sont très similaires puisqu'elles commencent dans la classe la plus vertueuse en 1980, traversent la position intermédiaire dans les années 90 pendant la récession et terminent dans la classe la plus vertueuse à la fin de la période (voir figure 1.4).

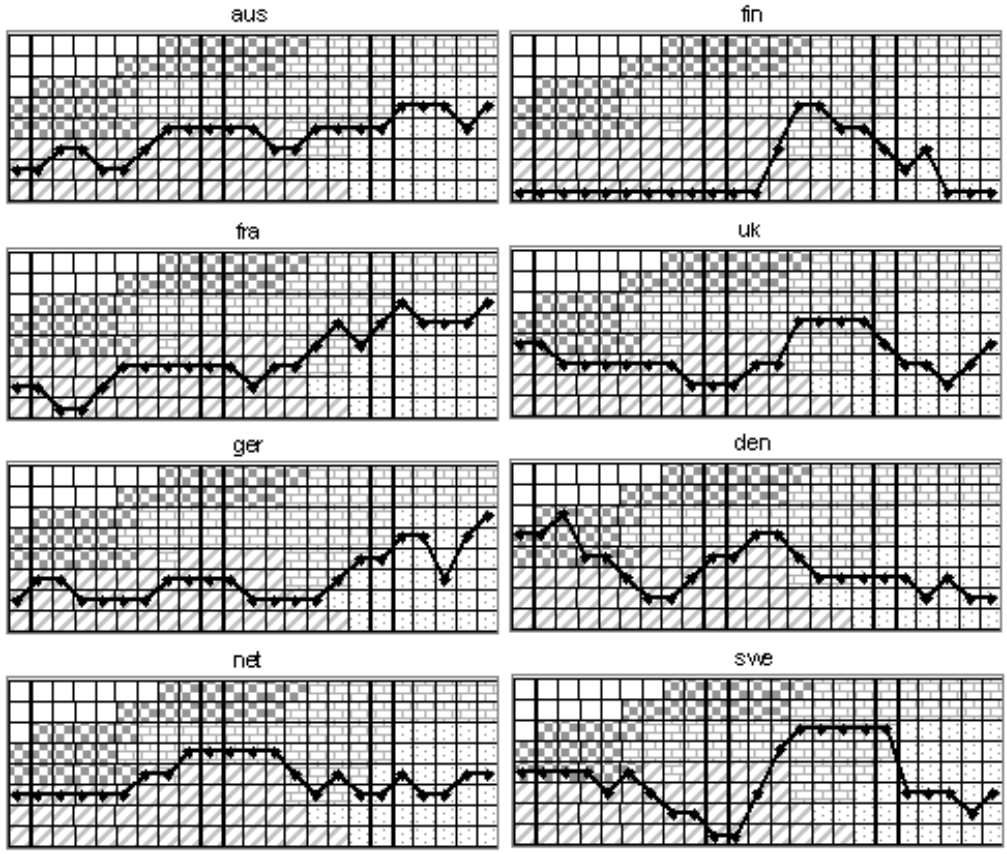


Fig. 1.4: Trajectoires de l'Autriche, de la France, de l'Allemagne, des Pays-Bas, du Royaume-Uni, du Danemark, de la Finlande et de la Suède

Les trajectoires de l'Italie, de la Belgique et de la Grèce sont opposées à celle du Luxembourg puisqu'elles sont restées très proches de la position la moins vertueuse durant toute la période (voir figure 1.5).

Deux trajectoires méritent une attention particulière, il s'agit de celles du Portugal et de l'Irlande (voir figure 1.6). Alors que ces deux pays commencent dans la classe la moins vertueuse en 1980, ils parviennent à passer dans la classe intermédiaire dans la moitié des années 90 (alors qu'à cette période, la plupart des pays européens voit leur situation s'aggraver suite à la récession en Europe) et terminent finalement dans la classe des pays les plus vertueux. Notre étude illustre

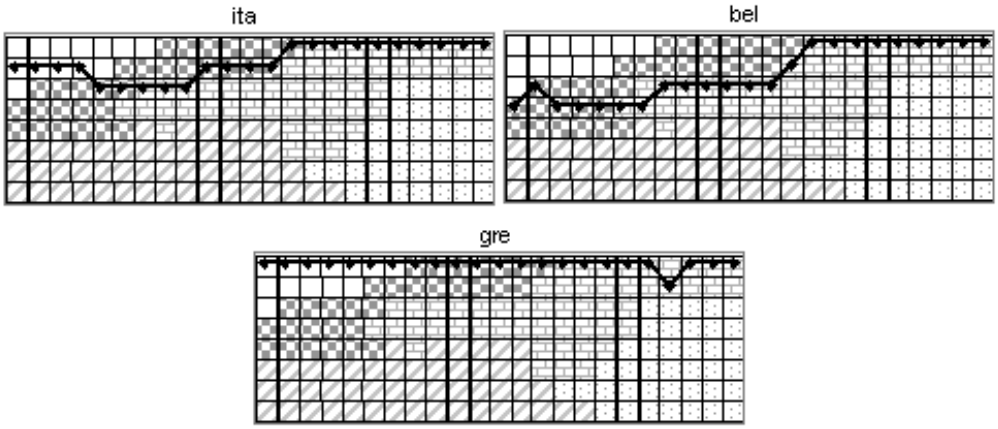


Fig. 1.5: Trajectoires de l'Italie, de la Belgique et de la Grèce

clairement la transition exceptionnelle de ces pays.

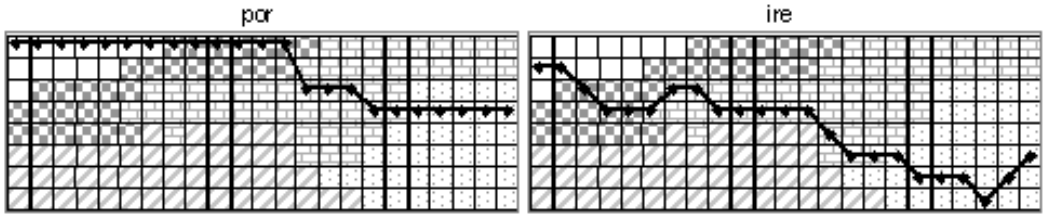


Fig. 1.6: Trajectoires du Portugal et de l'Irlande

La trajectoire de l'Espagne est très fluctuante, illustrant les difficultés pour cette économie à converger vers les normes de Maastricht (voir figure 1.7).

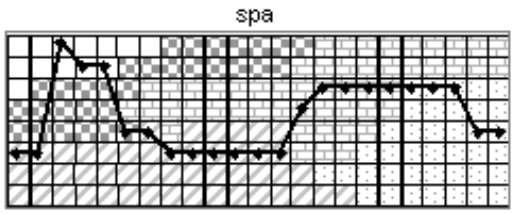


Fig. 1.7: Trajectoire de l'Espagne

2. ETUDE DES DYNAMIQUES DE GROUPES

2.1 Introduction

On va, dans cette partie s'intéresser à l'étude des dynamiques de groupes de données projetées sur une carte de Kohonen pour l'observation des comportements des gérants de SICAV "France" (c'est-à-dire constituées de plus de 80 pourcents d'actions françaises). Les performances des cartes de Kohonen pour traiter de telles données ont été démontrée par M. Verleysen, B Maillet et T. Kohonen par exemple. Pour pouvoir étudier la dynamique de groupe l'idée a été la suivante :

- projections des données à toutes les dates sur une carte de Kohonen
- analyse de la carte de Kohonen
- analyse de la fréquentation de la carte de Kohonen date par date

Dans un premier temps, on présentera les données que nous avons étudiées et les traitements statistiques que nous leur avons fait subir, puis nous présenterons la carte de Kohonen qui servira de base à la suite des traitements, enfin la méthode d'analyse de dynamique de groupe sera présentée ainsi que quelques uns des resultats jugés les plus intéressants.

2.2 Base de données et traitements préliminaires

La base est initialement composée des valeurs quotidiennes de 83 sicavs $V_t0(sicav)$. Nous avons choisit de travailler sur les rendements à 15 jours ($R_t(sicav) = V_t(sicav)/V_t(sicav) - 1$. Ce choix résulte de plusieurs tests et s'est avéré être un bon compromis entre de bons résultats de régressions à venir et des séries relativement peu lissées. En plus des rendements sur les sicavs on dispose des rendements d'indices : le CAC40, le SBF80, le second marché (SM), et le marché des nouvelles technologies (IT). Dans un premier temps, on va dé-corréler les indices (fortement corrélés

entre eux pour certains) par des régressions linéaires et un travail sur les résidus : Le modèle (donnant des coefficients significatifs à 95 pour cent) se résume par le système S_1 d'équations suivant :

$$R_t(SM) = \alpha_1 R_t(CAC) + \varepsilon_t(SM)$$
$$R_t(SBF80) = \alpha_2 R_t(CAC) + \beta_2 \varepsilon_t(SM) +_t (SBF80)$$
$$R_t(IT) = \alpha_3 R_t(CAC) + \beta_3 \varepsilon_2(SM) + \varepsilon_t(IT).$$

On travaillera alors, en régression sur les indices dé-correlés :

$$R_t(CAC), R_t(NR), \varepsilon_t(SM), \varepsilon_t(SBF80), \varepsilon_t(IT),$$

et on caractérisera les Sicavs par les coefficients de régressions sur les indices dé-corrélés les régressions étant effectuées sur des périodes d'environ 2 mois (61 jours centrés sur la date voulue) :

$$R_t(si) = a_t(si)R_t(CAC), +b_t(si)R_t(NR) + c_t(si)\varepsilon_t(SM) + d_t(si)\varepsilon_t(IT)$$

dont on gardera les coefficients significatifs à 95 pourcents. On ne gardera que les équations pour lesquelles le R^2 est supérieur à $0,6$. Ensuite on calculera le poids des indices par inversion du système S_1 sur (a, b, c, d) Enfin, comme on interprétera les poids comme des pourcentages de placement sur les différents indices, on ne conservera finalement que les poids positifs, les autres étant remis à 0 (voir Verleysen) et on normera les vecteurs pour que la somme des poids soit 1.

2.3 résultats de la projection sur une carte de Kohonen

On a projeté l'ensemble des poids sur une carte de Kohonen, cela représente au total 64773 vecteurs de dimension 4, en réalité en dimension 3 (somme des composantes égale à 1) et qui se projettent bien en dimension 2. Un vecteur poids correspondant à position stratégique d'une Sicav a un temps donné. La carte choisie est une carte $(7, 7)$ qui représente un compromis entre une projection fine des données (carte assez grande) et une fréquentation assez significative date par date pour les traitements à venir.

En résumé, on note que les strategies fortement axées sur le CAC40 se trouvent dans le coin supérieur gauche, celles axées sur le second marché sont dans le coin inférieur gauche, celles axées sur le SBF80, dans le coin inférieur droit, et les nouvelles technologies, assez peu représentées, sont au centre.

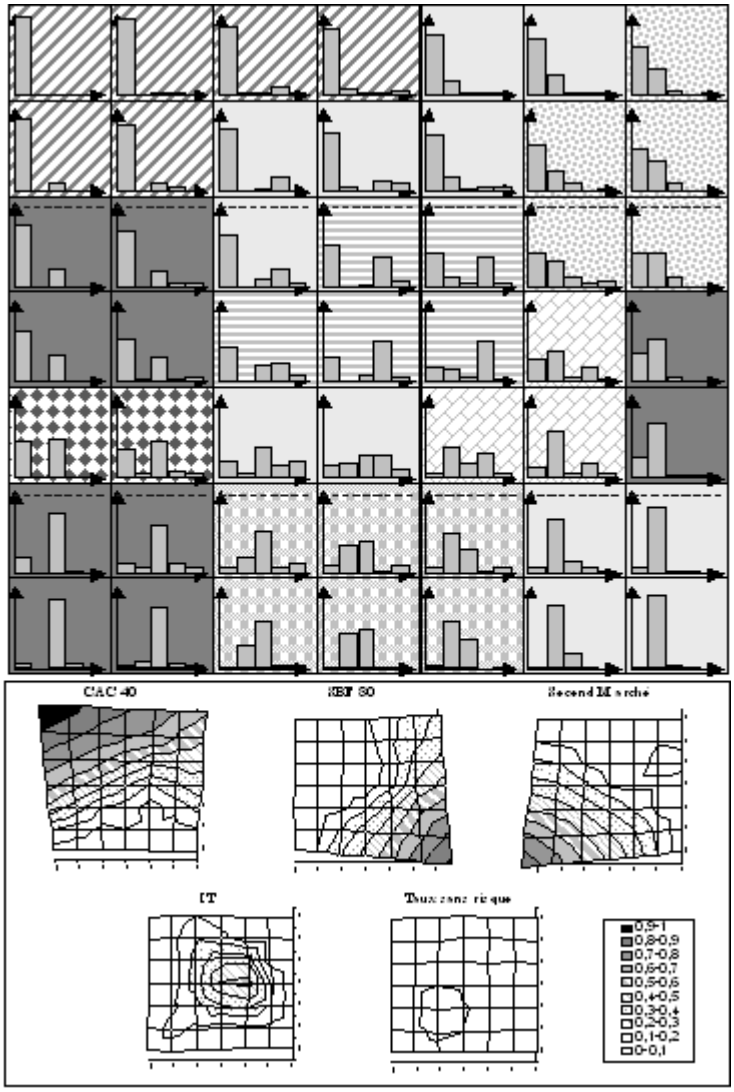


Fig. 2.1: Projection des différentes "stratégies" sur une carte $(7, 7)$

2.4 Analyse de dynamique de groupe

2.4.1 Individus et variables

Pour l'étude de la dynamique de groupe on a choisi de caractériser chaque date (en réalité chaque paquet de trois dates consécutives pour augmenter la signification) par une matrice $F(t) \in M(7, 7)$ représentant la fréquentation de la carte de Kohonen à la date t. Pratiquement $F(t)_{i,j}$

vaut le nombre de sicav à la date t situés dans la case (i, j) de la carte divisé par le nombre total de sicav étudiées à la date t (ce nombre peut varier d'une date à l'autre étant donné qu'on n'a gardé que les poids issus de régressions dont le R^2 était supérieur à $0,6$.

En résumé les individus seront désormais les dates (273 dates formées de paquets de 3 jours consécutifs) et les variables les matrices de fréquentation.

2.4.2 Distance entre les matrices de fréquentation

Du fait de la structure topologique de la carte de Kohonen, une distance "classique" entre les matrices de fréquentation ne sera pas très pertinente car elle ne rendra pas compte de la proximité entre les cellules de la carte (voir figure 2.2) .

Supposons qu'on ait les trios matrices de fréquentation suivante

$$A=\begin{pmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}, B=\begin{pmatrix}0&1&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}, C=\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&1\end{pmatrix}$$

- en utilisant la distance euclidienne $d(A,B) = d(A,C) = d(B,C) = \sqrt{2}$
- toute distance « classique » donnera $d(A,B) = d(A,C) = d(B,C)$
- dans le cas de matrices de fréquentation on s'attend a : $d(A,B) < d(A,C)$

Fig. 2.2: Encadré sur les distances

Pour tenir compte des effets de proximité entre les cellules des matrices de fréquentation, on considérera la fréquentation comme une surface, les données brutes correspondant à un histogramme qu'on va lisser par des noyaux gaussiens : on passe ainsi des matrices de fréquentation $(F(t))$ à des "nappes" de fréquentation (f_t) par :

$$f_t(x,y) = \sum_{i,j} F_{i,j}(t) K((x,y),(i,j),\sigma^2 Id)$$

avec :

$$K((x,y),(i,j),\sigma^2 Id) = \frac{1}{2\pi\sigma^2} exp(-\frac{(x-i)^2+(y-j)^2}{2\sigma^2})$$

Les distances entre les nappes de fréquentation sont alors très simples à calculer :

$$d(f_t, f_{t'}) = \int ((f_t - f_{t'})^2$$

$$= \sum_{i,j,i',j'} (F_{i,j}(t) - F_{i,j}(t'))(F_{i',j'}(t) - F_{i',j'}(t'))exp(-\frac{(i-i')^2 + (j-j')^2}{4\sigma^2})$$

Le choix du paramêtre σ est fondamental : si $\sigma \to 0$ on va tendre vers la distance euclidienne, si σ est trop grand, on aura des distances très faibles entre toutes les cartes.

On a choisit $\sigma = 1/2$ en se référant au cas ou la fréquentation est uniforme dans la nappe et en choisissant le paramètre qui rend la nappe la plus "uniforme".

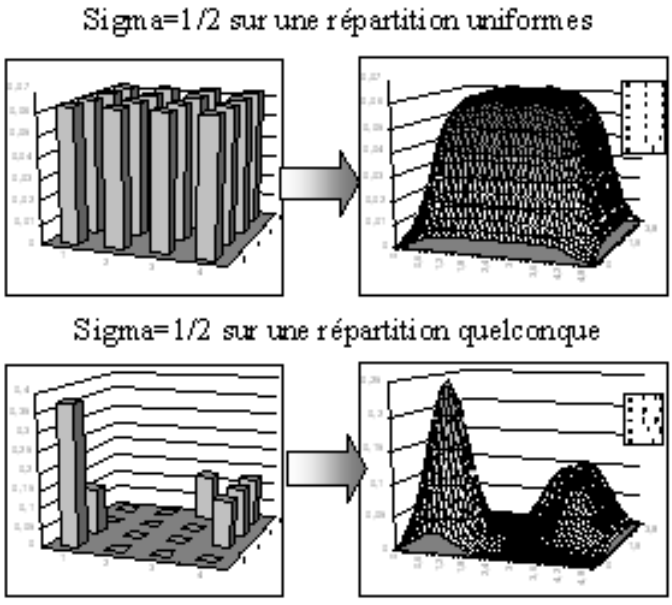


Fig. 2.3: Le choix de $\sigma = 0.5$

2.4.3 Classification

Une fois définis ce qu'étaient les individus (temps), les variables (matrices de fréquentation) et les distances entre variables, on va procéder à

une classification des temps en période d'évolution par une classification hiérarchique par la distance minimum avec un indicateur de distance intra classe calculé comme en partie 2 (classification). Dans le cas où les données sont aussi particulières, les tests sur les ruptures de distance intra classe sont totalement non adaptés et on se contentera de choisir un nombre de classes correspondant à la plus grande valeur de rupture, soit 29 classes.

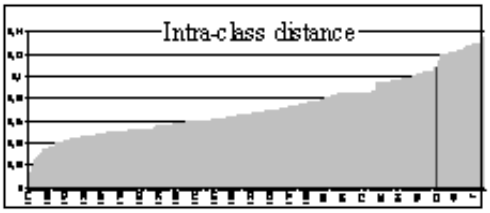


Fig. 2.4: nombre de classes

2.5 Résultats

2.5.1 Résultats généraux sur l'ensemble de la période

Considérons l'interprétation du premier carré à gauche (date 1/03/99), on constate une dispersion des gérants (pas de zone foncée) avec toutefois trois pôles sur le CAC 40, sur le MC et sur le SBF. Dans le carré suivant en bas, on observe une polarisation des gérants sur le CAC 40. Le premier carré renvoie à la période 11/03/1999 à 11/10/1999, soit la période située entre les deux premières lignes verticales. Le carré en-dessous caractérise la période suivante, de même pour les troisième et quatrième carrés. Les dates indiquées au-dessus des flèches correspondent aux changements de colonnes.

Nous pouvons suivre l'évolution globale des fonds, à travers les classes de Kohonen. Nous pouvons distinguer dix périodes :

- 1) Mars 99- octobre 99: Au début de la période, peu de ruptures sont à signaler. On observe la présence de quatre pôles : CAC 40, un mélange CAC40 MC, MC et SBF (plutôt mixe de CAC 40 et SBF 80).

- 2) Décembre 99- mai 2000 : On observe des dispersions accrue des profils. La figure 9 détaille les changements constatés durant cette

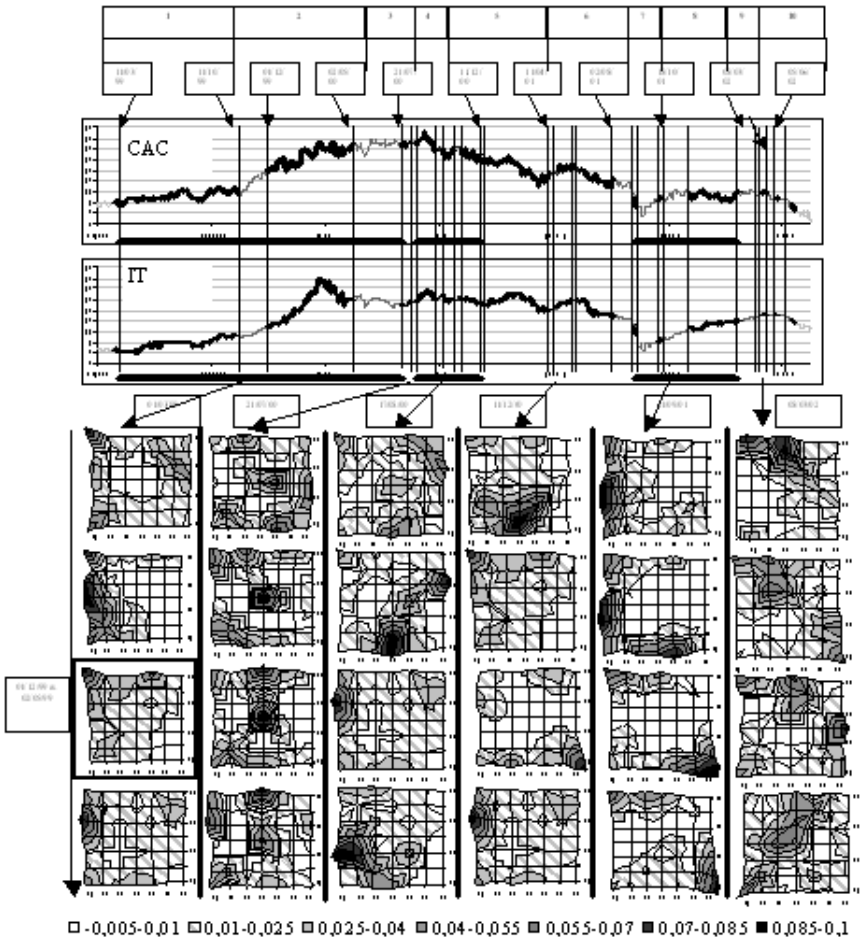


Fig. 2.5: évolution sur toute la période

période.

- 3) Juillet-août 2000 : Les rendements des fonds sont fortement liés aux cours de l'IT et du CAC 40. La chute brutale sur la Nouvelle Technologie, accompagnée d'un choc de liquidité explique sans doute cette forte corrélation. La valeur liquidative des fonds a fortement baissé.

- 4) Août- novembre 2000 : Les fonds cherchent des solutions de repli : surtout vers les pôles SBF 80 et MC, et peu vers le CAC 40.

- 5) Décembre 2000-Avril 2001 : On constate une forte hétérogénéité des styles en début de période puis l'apparition d'un pôle important

SBF 80 et de pôles secondaires CAC 40, SBF et MC. La zone du centre est délaissée : les positions sur l'IT ont été dénouées.

- 6) Mai- Août 2001 : Il existe une forte dispersion, le pôle SBF 80 est un peu abandonné au profit des pôles CAC 40 et SBF.
- 7) Septembre 2001 : On constate un repli brutal et collectif vers le CAC 40.
- 8) Octobre- Mars 2002 : Le pôle SBF 80 constitue un pôle de concentration très important, le segment MC est délaissé.
- 9) Mars 2002 : Les fonds se replient vers le CAC 40, le SBF 80 n'attire plus.
- 10) Avril- Juin 2002 : On observe une forte dispersion des profils, le retour du pôle MC, la disparition du pôle SBF 80, la réapparition de fonds dans la zone du centre.

2.5.2 Détail de la période " bulle internet "

La figure ci-dessus indique toutes les cartes de fréquentation pour la troisième classe, à savoir celle qui débute le 01/12/99 et finit le 02/05/00, soit 96 jours ouvrés. Dans la mesure où nous avons travaillé sur 3 jours, nous obtenons 35 individus. Cette période correspond à la bulle Internet, marquée par une croissance très forte des cours des indices IT, NM et CAC 40.

Nous pouvons suivre au jour le jour les transformations de la carte des fréquentations. La lecture du graphe se fait de gauche à droite, puis de haut en bas. Nous constatons déjà que, au sein de cette classe, les évolutions sont continues. Il est possible de distinguer trois sous-périodes :

- 1) 01/12/99- 13/01/00 : Cette période est caractérisée car un engouement fort pour le CAC 40 et le SBF ainsi que l'absence du pôle SBF 80. On observe aussi que quelques fonds s'aventurent déjà dans la zone du centre.
- 2) 18/01/00- 29/03/00 : Cette phase correspond à l'engouement pour la Nouvelle Technologie. On constate en effet que la zone du centre, correspondant au pôle IT, s'avère très attractive. Le CAC 40 et le MC attirent aussi de nombreux fonds.

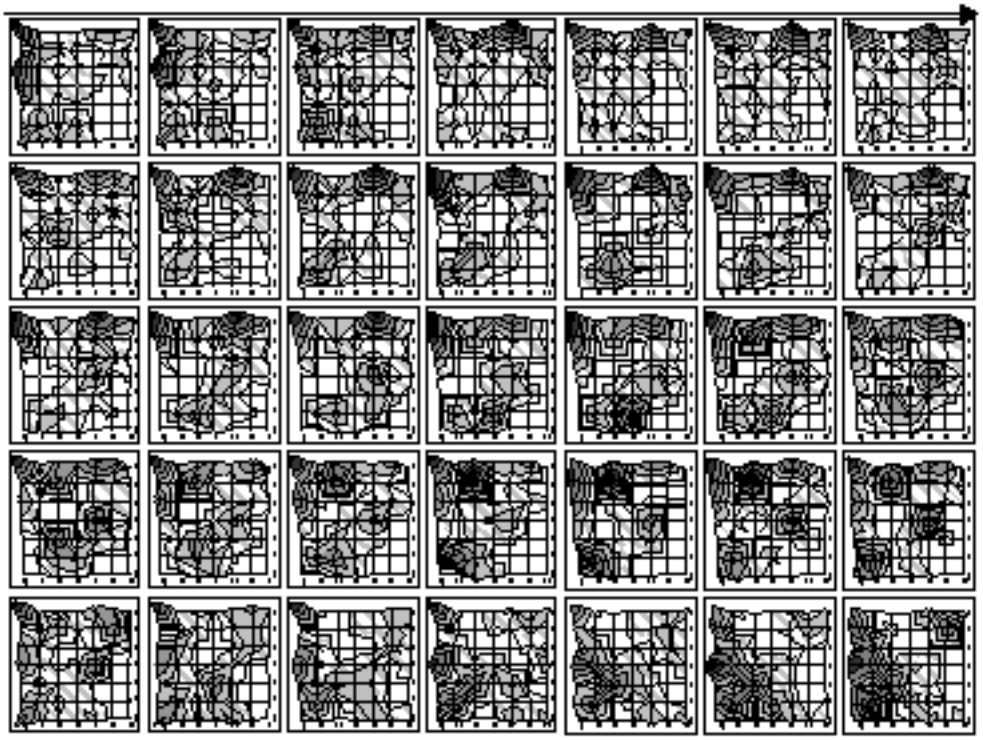


Fig. 2.6: Evolution collective détaillée 01/12/99-02/05/00

- 3) 03/04/00- 02/05/00 : Le pôle IT perd peu à peu de son pouvoir attractif, au profit du CAC 40 et du MC. Le pôle SBF 80 est toujours absent.

Part IV

BIBLIOGRAPHIE

K MEANS

[KMN1] Forgy E. W. **(1965)** *"Cluster analysis of multivariate data: efficiency vs interpretability of classifications"* Biometrics, 21, pp 768 − 769
[KMN2] Hartigan J. A. ,Wong M. A **(1979)** *"A K-means clustering algorithm"* Applied Statistics, 28, pp 100108
[KMN3] Lloyd S. P.**(1982)** *"Least squares quantization in PCM"* IEEE Transactions on Information Theory 28, pp 128137
[KMN4] MacQueen J. **(1967)** *"Some methods for classification and analysis of multivariate observations"* In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp 281297
[KMN5] Miranda M. I.**(1999)** *"Clustering methods and algorithms"* http://www.cse.iitb.ac.in/dbms/Data/Courses/CS632/1999/clustering/dbms.htm
[KMN6] Moore A. *"K-means and Hierarchical Clustering - Tutorial Slides"* http://www-2.cs.cmu.edu/ awm/tutorials/kmeans.html
[KMN7] Zaïane O. R. *"Principles of Knowledge Discovery in Databases, Chapter 8: Data Clustering"* http://www.cs.ualberta.ca/ zaiane/courses/cmput690/slides/Chapter8/index.html

CLASSIFICATION HIERARCHIQUE AVEC LA DISTANCE DU MINIMUM

[HIE1] Anderberg, M. R.**(1973)** *"Cluster Analysis for Applications"* Academic Press: New York.
[HIE2] Benzecri, J.P. **(1973)** *"L'analyse des données"* T1 : La taxinomie. Dunod.
[HIE3] Borgatti S. P. **(1994)** *"How to explain hierarchical clustering"* Connections, 17 pp 78 − 80 http://www.analytictech.com/networks/hiclus.htm
[HIE4] Carroll J.D.**(1995)** *"Minimax Length Links of a dissimilarity matrix and minimum spanning trees"* Psychometrika, pp 371 − 374
[HIE5] Carroll J.D., De Soete G.**(1996)** *"Tree and other network models for representing proximity data"* Clustering and Classification River Edge, NJ: World Scientific, pp 157 − 197
[HIE6] Cormack, R. M. **(1971)** *"A review of classification"* Journal of the Royal Statistical Society $A134$, pp 321 − 367
[HIE7] D'andrade R.**(1978)** *"U-Statistic Hierarchical Clustering"* Psychometrika 4, pp 58 − 67

[HIE8] Diday, E., J. Lemaire, J. Pouget, and F. Testu **(1982)** *"Elements d'analyse de données"* Dunod, Paris.
[HIE9] Everitt, B. **(1974)** *"Cluster Analysis"* London: Heinemann Educ. Books.
[HIE10] Gordon, A. D. **(1999)** *"Classification. Second Edition"* London: Chapman and Hall / CRC
[HIE11] Hartigan, J. A. **(1975)** *"Clustering Algorithms."* New York: Wiley.
[HIE12] Hastié T., Tibshirani R., Walter G.**(2000)** *"Estimating the Number of Data Cluster via the Gap Statistic"* www-stat.stanford.edu/ tibs/ftp/gap.ps
[HIE13] Johnson S. C.**(1967)** *"Hierarchical Clustering Schemes"* Psychometrika 2, pp 241 − 254
[HIE14] Lebart, L., A. Morineau, and M. Piron. **(1995)** *"Statistique exploratoire multidimensionnelle"* Dunod, Paris.
[HIE15] Murtagh, F. **(1985)** *"Multidimensional Clustering Algorithms"* in COMPSTAT Lectures 4. Wuerzburg:Physica-Verlag
[HIE16] Roux, M. **(1985)** *"Algorithmes de classification"* Masson.
[HIE17] Sokal, R. R., and Sneath P. H. A.**(1963)** *"Principles of numerical taxonomy"* Freeman and Co., San-Francisco.

[HIE18] Gower, J.C., and Ross, G.J.s. **(1969)** *"Minimum Spanning Tree and Single Linkage Analysis"* Applied Statistics Vol 18 N1 pp54−64

CLASSIFICATION CONNEXITE ET IMAGES

[PAT1] Baraldi, A. Parmiggiani, F.**(1995)** *"A neural network for unsupervised categorization of multivaluedinput patterns: an application to satellite image clustering Geoscience and Remote Sensing"* IEEE Transactions on neural network 33, pp 305 − 316
[PAT2] Chulhee L., Shin H, Terence A. Ketter B,**(1998)** *"Unsupervised connectivity-based thresholding segmentation of midsagittal brain MR images"* Computers in Biology and Medicine 28, pp 309 − 338
[PAT3] S Lee, MM Crawford **(2005)** *"Unsupervised Multistage Image Classification Using Hierarchical Clustering With a Bayesian"* IEEE Transactions on Image Processing (à paraître)

METHODES A BASE DE DENSITE

[DST1] Bicego M., Cristani M., Fusiello A., Murino V.**(2003)** *"Watershed-based unsupervised clustering"* http://profs.sci.univr.it/ bicego/bicego$_m$*urino*$_e$*mmcvpr*03.*pdf*.
[DST2] Meyer, F.**(1994)** *"Topographic distance and watershed lines"* Signal Processing 38, pp 113125
[DST3] MICHALIS K., TITSIAS., ARISTIDIS C LIKA S**(2001)** *"Shared Kernel Models for Class Conditional Density Estimation."* IEE Transaction on Neural Network 12, pp 987 − 997
[DST4] Roerdink, J.B.T.M., Meijster, A.**(2000)** *"The watershed transform: Definitions, algorithms and parallelization strategies"* Fundamenta Informaticae 41, pp 187228
[DST5] SAIN S., SCOTT W. **(1996)** *"On Locally Adaptive Density Estimation"* Journal of the American Statistical Association N436 ftp://ftp.stat.rice.edu/pub/scottdw/Tech.Reps/adapt.ps.
[DST6] STUETZLE W.**(2003)** *"Estimation of the cluster tree of a density by analysing the minimal spanning tree of a sample"* Journal of the American Statistical Association http://www.stat.washington.edu/wxs/Learning-papers/mst.pdf
[DST7] WISHART D.**(1969)** *"Mode Analysis : A Generalization of Nearest Neighbor which Reduce Chaining Effect"* Numerical Taxinomy, pp 282 − 311

[DST8] Ester M., Kriegel H.P., Sander J., Xu X.**(1996)** *"A Density Based Algoriyhm for Discovering Clusters in Large Spatial Database with Noise"* proceeding of the 2^{the} conference on knowledge discovery and data analysis KDD

CLASSIFICATION SPECTRALE

[SPC1] Bach F. R. and Jordan M. I. **(2004)** *"Blind one-microphone speech separation: A spectral learning approach"* http://cmm.ensmp.fr/ bach/fbach$_n$*ips*$_2$004$_s$*peech.pdf*
[SPC2] F. R. Bach and M. I. Jordan. In S. Thrun, L. Saul, and B. Schoelkopf **(2004)** *"Learning spectral clustering"* Advances in Neural Information Processing Systems
[SPC3] A. Y. Ng, M. I. Jordan, and Y. Weiss. In T. Dietterich, S. Becker and Z. Ghahramani **(2002)** *"On spectral clustering: Analysis and an algorithm"* Advances in Neural Information Processing Systems
[SPC4] E. P. Xing and M. I. Jordan **(2003)** *"On semidefinite relaxation*

for normalized k-cut and connections to spectral clustering" Technical Reports
[SPC5] Luh Yen, Denis Vanvyve, Fabien Wouters, Francois Fouss, Michel Verleysen Marco Saerens **(2005)** *"Clustering using a random walk based distance measure"* ESANN'2005 proceedings - European Symposium on Artificial Neural Networks
[SPC6] Zhukov L. **(2004)** *"Spectral Clustering of Large Advertiser Datasets"* Technical Report http://www.gg.caltech.edu/ zhukov/papers/bipartite-spectral-clustering-report.pdf

GROWING NEURAL GAS

[GNG1] Martinetz T. M. and Schulten K. J. **(1991)** *"A neural-gas network learns topologies"* In Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., editors, Artificial Neural Networks, North-Holland, Amsterdam, pp 397 – 402
[GNG2] Martinetz T. M **(1993)** *"Competitive Hebbian learning rule forms perfectly topology preserving maps"* ICANN93: International Conference on Artificial Neural Networks, Springer, Amsterdam, pp 427 – 434
[GNG3] Fritzke, B. homepage and demo : http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/GNG.html
[GNG4] Fritzke, B. **(1994)** *"Fast learning with incremental RBF networks"* Neural Processing Letters 1, pp 2 – 5
[GNG5] Fritzke, B. **(1997)** *"A self-organizing network that can follow non-stationary distributions"* Proc. of the International Conference on Artificial Neural Networks '97 , pp 613 – 618

CARTES DE KOHONEN

[SOM1] Cottrell, M., Fort, J.C. **(1986)** *"A stochastic model of retinotopy : a self-organizing process"* Biological Cybernetics 53, pp 405411
[SOM2] T. Kohonen **(1982)** *"Self-Organized formation of topologically correct feature maps"* Biological Cybernetics 43, pp 59 – 69.
[SOM3] Kohonen, T. **(1984)** *"Cybernetic Systems: Recognition, Learning, Self-Organization"* In: Caianiello, E.R., Musso, G. (Eds.), Research Studies Press, Ltd., Letchworth, Herfordshire, UK

[SOM4] Kohonen, T.**(1989)** *"Self-Organization and Associative Memory"* 2nd Edition. Springer, Berlin.
[SOM5] Kohonen, T.**(1995)** *"Self-Organizing Maps"* Springer, Berlin.
[SOM6] M. Dash and H. Liu **(2001)** *"Efficient hierarchical clustering algorithms using partially overlapping partitions"* Lecture Notes in Computer Science 2035, pp 495 – 507.
[SOM8] A. Ultsch **(1992)** *"Self-Organizing Neural Networks for Visualization and Classification"* Proc. Conf. Soc. for Information and Classification, Dortmund (Germany), April 1992.
[SOM9] A. Ultsch **(2003)** *"Maps for the Visualization of high-dimensional Data Spaces"* In Proc. WSOM03, Kyushu (Japan) pp 225 – 230.
[SOM10] A. Ultsch **(2003)** *"U*-Matrix: a Tool to visualize Clusters in high dimensional Data"* In Research report Dept. of Mathematics and Computer Science, University of Marburg (Germany), No. 36.
[SOM11] A. Ultsch and H.P. Siemon **(1990)** *"Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis"* In Proc. Intern. Neural Networks Conf. (INNC90), Dortrecht (Netherlands), Kluwer Academic Press, Paris, pp 305 – 308
[SOM12] A. Ultsch and C. Vetter **(1994)** *"Self-Organizing-Feature-Maps versus statistical clustering methods: a benchmark"* FG Neuroinformatik and Kuenstliche Intelligenz, University of Marburg, Research Report 0994.
[SOM13] Moutarde F. and Ultsch A.**(2005)** *"U*F clustering: a nw performant "cluster-mining" method based on self organizing maps"* In Proc. Of WSOM'05 conference, Paris
[SOM14] J. Vesanto et al. **(1999)** *"Self-organizing map in Matlab: the SOM toolbox"* the Matlab DSP Conference, Espoo, Finland, November 1999, pp 35 – 40
[SOM15] J. Vesanto **(1999)** *"SOM-based data visualization methods"* Intelligent Data Analysis $Vol3$
[SOM16] J. Vesanto **(2000)** *"Using SOM in data mining"* Licentiate thesis, Helsinki University of Technology.
[SOM17] J. Vesanto and E. Alhoniemi **(2000)** *"Clustering of the Self-Organizing Map"* IEEE Transactions on Neural Networks" 11, pp 586 – 600 http://lib.tkk.fi/Diss/2002/isbn951226093X/article4.pdf
[SOM18] M. Cottrell E. De Bodt **(1996)** *"CA Kohonen Map Representation to Avoid Misleading Interpretations"* ESANN'1996 proceedings - European Symposium on Artificial Neural Networks pp. 103 – 110

STATISTIQUES SUR LES LONGUEURS DE LIAISON DU MST

[MST1] D. Aldous, J.M. Steel **(1992)** *"Asymptotics for Euclidian Minimal Spanning Tree on Random Points"* Probability Theory and Related Fields 92, pp 247 − 258
[MST2] K.S. Alexander **(1996)** *"The RSW Theorem for Continuum Percolation And the CLT For Euclidian Minimal Spanning Tree"* The Annals of Applied Probability 6, pp 466 − 494
[MST3] J. Beardwood, J.H. Hammersley **(1959)** *"The Shortest Path Through Many Points"* Proc. Cambridge Philos. Soc 55, pp 299 − 327
[MST4] H. Kesten, S. Lee **(1996)** *"The Central Limit Theorem for weighted Minimal Spanning Tree"* The Annals of Applied Probability 6, pp 495 − 527
[MST5] S.Lee **(1997)** *"The Central Limit Theorem for Euclidian Minimal Spanning Tree I"* The Annals of Applied Probability 7, pp 996−1020
[MST6] M.D. Penrose **(1997)** *"The Longest Edge of the Random Minimal Spanning Tree"* The Annals of Applied Probability 7, pp 340 − 361
[MST7] M.D. Penrose**(1998)** *"Extremes for Minimal Spanning Tree on Normaly Distributed Points"* Advanced on Applied Probability 30, pp 628 − 639
[MST8] M.D. Penrose **(1996)** *"The Random Minimal Spanning Tree in Hight Dimension"* The Annals of Applied Probability 24, pp 1903−1925
[MST9] M.D. Penrose **(2003)** *"Weak Law of Large Number in Geometric Probability"* The Annals of Applied Probability 13, pp 277 − 303
[MST10] J.M. Steel **(1981)** *"Complete Convergence Of Short Paths And Karp's Algorithm for the TSP"* Mathematics of operation research 16, pp 375 − 377

ESTIMATION DE DENSITE

[DEN1] Abramson I. **(1982)** *"On Bandwidth Variation in Kernel Estimates A Square Root Law"* The Annals of Statistics 10, pp 1217 − 1223
[DEN2] Bartlett M.S. **(1963)** *"Statistical Estimation of Density Functions"* Sankhya Series A 25, pp 25
[DEN3] Duin R.P.W. **(1976)** *"On the choice of smoothing parameters of parzen estimators of probability density functions"* IEEE Transaction on Computers 25, pp 1175 − 1179

[DEN4] Habbema J.D.F., Herman J., Van den Broek K.**(1974)** *"a stepwise discrimination analysis program using density estimation"* COMPSTAT'74 proceeding in computational statistics

[DEN5] Parzen E. **(1979)** *"Nonparametric Statistical Data Modeling with discussion"* Journal of the American Statistical Association 85, pp 66−72

[DEN6] Parzen E. **(1962)** *"On estimating of a probability density and modes"* annals of mathematical statistics 35, pp 1065 − 1076

[DEN7] Sain S.R. **(1996)** *"Adaptive Kernel Density Estimation"* Unpublished dissertation De partment of Statistics Rice University

[DEN8] Sain S.R. and David W.S. **(1996)** *"On Locally Adaptive Density Estimation"* Journal of the American Statistical Association 91, pp 1525 − 1534

PROJECTION

[CRV1] Akkucuk, U., and Carroll, J. D. **(2004)** *"Mapping of nonlinear manifolds: ISOMAP and a version of PARAMAP"* In D. Banks, L. House, F. R. McMorris, P. Arabie, W. Gaul (Eds.), Classification, Clustering and Data Mining Applications. Springer: Berlin. Proceedings of the meeting of IFCS/CSNA

[CRV2] Lee J.A. , Lendasse A., Verleyseny M. **(2002)** *"Curvilinear Distance Analysis versus Isomap"* ESANN'2002 proceedings - European Symposium on Artificial Neural Networks Bruges (Belgium) pp 185−192

[CRV3] Tenenbaum J. B., de Silva V. and Langford J. C.**(2000)** *"A Global Geometric Framework for Nonlinear Dimensionality Reduction"* Science 290, pp 2319 − 2323

[CRV4] Zhi-Hua Zhou, Yang Yu**()** *"Ensembling Local Learners Through Multimodal Perturbation"* S-ISOMAP ieee transactions on systems

DIMENSION

[DIM1] R. Badii and A. Politi **(1985)** *"Statistical description of chaotic attractors: the dimension function"* J. Stat. Phys 40, pp 725

[DIM2] F. Camastra and A. Vinciarelli **(2002)** *"Estimating intrinsic dimension of data with a fractal-based approach"* IEEE Transactions on Pattern Analysis and Machine Intelligence

[DIM3] P. Grassberger and I. Procaccia**(1983)** *"Measuring the strangeness of strange attractors"* Physica *vol.D*9, pp 189208

[DIM4] P. Grassberger **(1985)** *"Generalizations of the Hausdorff dimension of fractal measures"* Phys. Lett. A 107 pp 101 – 105
[DIM5] J. Guckenheimer and G. Buzyna **(1983)** *"Dimension measurements for geostrophic turbulence"* Phys. Rev. Lett 51, pp 1438 – 1441
[DIM6] Balazs Kegl **(2002)** *"Intrinsic Dimension Estimation Using Packing Numbers"* www.iro.umontreal.ca/ kegl/ research/publications/kegl02.ps
[DIM7] B. Mandelbrot **(1977)** *"Fractal Geometry of Nature"* W.H. Freeman, New York
[DIM8] F. Takens **(1983)** *"Invariants related to dimension and entropy"* Atas do 130 (Colokio Brasiliero do Matematica, Rio de Janeiro)
[DIM9] J Theiler **(1990)** *"Estimating fractal dimension"* J. Opt. Soc. Am. A 7, pp 1055−1073 http://nis-www.lanl.gov/ jt/Papers/est-fractal-dim.pdf
[DIM10] Angeline Wong Leejay Wu Phillip B. Gibbons**(2003)** *"Fast Estimation of Fractal Dimension and Correlation Integral on Stream Data"* www.db.cs.cmu.edu/Pubs/Lib/ fracKDD03tugofwar/$Tug_of_War_paper - KDD03.pdf$

MESURES DE PRESERVATION DE LA TOPOLOGIE

[TOP1] H-U Bauer and K. Pawelzic **(1992)** *"Quantifying the neigborhood preservation of self-organizing maps"* ieee transaction on Neural Network 3, pp 570 – 579
[TOP2] E. de Bodt, M. Cottrell and M. Verleysen **(2002)** *"Statistical tools to asses the reliability of self organizing maps"* Neural Network 15, pp 967 – 978
[TOP3] P. Demartines **(1994)** *"Analyse des données par réseaux de neurones auto-organisés"* These
[TOP4] T. Villmann, R. Der, M. Herrmann and T. Martinez **(1997)** *"Topology Preservation in Self-Organizing Feature Maps : Exact Definition and Measurement"* ieee transaction on neural networks $vol.8N2$, pp 256 – 266
[TOP5] S. Zrehen **(1993)** *"Analysing Kohonen maps with geometry"* Proceedings of ICANN , pp 609 – 612
[TOP6] G.J. Goodhill, T. Sejnowski **(1996)** *"Quantifying neighbourhood preservation in topographic mapping"* Proceeding of the 3rd joint Symposium on Neural Computation 1, pp 61 – 82

ETUDES DES DYNAMIQUES INDIVIDUELLES

[DYI1]Akarçay-Gürbüz A., Perraudin C. **(2002)**,*"Comment situer l'économie de la Turquie parmi les économies de l'UE? Une analyse exploratoire"*, it Proc. ACSEG 2002, Boulogne Sur Mer, EJESS 2003.
[DYI2]Cottrell M. **(2003)**, *"Some Other Applications of the SOM algorithm: how to use the Kohonen algorithm for forecasing"*, Prepub SAMOS 185.
[DYI3]Cottrell M., Rousset P. **(1997)**,*"The Kohonen algorithm: a powerful tool for analysing and representing multidimensional quantitative and qualitative data"*, itProc. IWANN'97, Lanzarote, June 1997, J. Mira, R. Moreno, J. Cabestany Eds, Lecture Notes in Computer Science, n 1240, Springer, p. 861-871.
[DYI4]Kohonen T. **(1995)**,*"Self-Organizing Maps"*, Springer Series in Information Sciences, Vol 30, Springer.
[DYI5]Letrémy P. **(2000)**, *"Notice d'installation et d'utilisation de programmes basés sur l'algorithme de Kohonen et dédiés à l'analyse des données*, Prepub SAMOS, 131.

ETUDES DES DYNAMIQUES DE GROUPE

[DYG1] Brown S, Goetzmann N. **(1997)** : *"Mutual fund styles"*, Journal of Financial Economics, 43, pp. $373-399$.
[DYG2] Cardon P., Lendasse A., Wertz V., De Bodt E., Verleysen M. **(2002)** : *"Classification de fonds communs d'investissement par cartes auto-organisées"*, ACSEG 2002.
[DYG3] Chan, Chen, Lakonishok **(2002)**: *"On Mutual Fund Investment Styles"*, Review of financial studies, 15, pp. $1407-1437$.
[DYG4] Cottrell M, De Bodt E., Pagès G. **(1997)** : *"Theoretical aspects of the Kohonen Algorithm"*, WSOM'97, Helsinki.
[DYG5] DiBartolomeo D., Witkowski E. **(1997)** : *"Mutual fund misclassification : Evidence based on style analysis"*, Financial Analysts Journal; 53(5), pp. $32-43$
[DYG6] Falkenstein E. G. **(1996)** : *"Preference s for stock characteristics as revealed by mutual fund portfolio holdings"*, Journal of finance, 51(1), mars, pp. $111-135$
[DYG7] Kim M., Shukla R., Tomas M. **(2000)** : *"Mutual fund objective misclassification"*, Journal of Economics and Business, 52, pp. $309-323$
[DYG8] Maillet B., Rousset P. **(2003)** : *"Classifying Hedge Funds using Kohonen Map"*, Forthcoming in Computational Economics, Series in

Adavnces in Computational Economics and Management Sciences.
[DYG9] Oja E., Kaski S. **(1999)** : *Kohonen Maps*, Elsevier.